

# Chapter 42

## The MODECLUS Procedure

### Chapter Table of Contents

---

<b>OVERVIEW</b>	2229
<b>GETTING STARTED</b>	2231
<b>SYNTAX</b>	2236
PROC MODECLUS Statement	2236
BY Statement	2243
FREQ Statement	2244
ID Statement	2244
VAR Statement	2244
<b>DETAILS</b>	2245
Density Estimation	2245
Clustering Methods	2248
Significance Tests	2251
Computational Resources	2257
Missing Values	2257
Output Data Sets	2258
Displayed Output	2260
ODS Table Names	2263
<b>EXAMPLES</b>	2264
Example 42.1 Cluster Analysis of Samples from Univariate Distributions	2264
Example 42.2 Cluster Analysis of Flying Mileages between Ten American Cities	2287
Example 42.3 Cluster Analysis with Significance Tests	2294
Example 42.4 Cluster Analysis: Hertzprung-Russell Plot	2300
Example 42.5 Using the TRACE Option when METHOD=6	2304
<b>REFERENCES</b>	2308



## Chapter 42

# The MODECLUS Procedure

---

### Overview

The MODECLUS procedure clusters observations in a SAS data set using any of several algorithms based on nonparametric density estimates. The data can be numeric coordinates or distances. PROC MODECLUS can perform approximate significance tests for the number of clusters and can hierarchically join nonsignificant clusters. The significance tests are empirically validated by simulations with sample sizes ranging from 20 to 2000.

PROC MODECLUS produces output data sets containing density estimates and cluster membership, various cluster statistics including approximate  $p$ -values, and a summary of the number of clusters generated by various algorithms, smoothing parameters, and significance levels.

Most clustering methods are biased toward finding clusters possessing certain characteristics related to size (number of members), shape, or dispersion. Methods based on the least-squares criterion (Sarle 1982), such as  $k$ -means and Ward's minimum variance method, tend to find clusters with roughly the same number of observations in each cluster. Average linkage (see Chapter 23, "The CLUSTER Procedure") is somewhat biased toward finding clusters of equal variance. Many clustering methods tend to produce compact, roughly hyperspherical clusters and are incapable of detecting clusters with highly elongated or irregular shapes. The methods with the least bias are those based on nonparametric density estimation (Silverman 1986, pp. 130–146; Scott 1992, pp. 125–190) such as density linkage (see Chapter 23, "The CLUSTER Procedure," Wong and Lane 1983, and Wong and Schaack 1982). The biases of many commonly used clustering methods are discussed in Chapter 8, "Introduction to Clustering Procedures."

PROC MODECLUS implements several clustering methods using nonparametric density estimation. Such clustering methods are referred to hereafter as *nonparametric clustering methods*. The methods in PROC MODECLUS are related to, but not identical to, methods developed by Gitman (1973), Huizinga (1978), Koontz and Fukunaga (1972a, 1972b), Koontz, Narendra, and Fukunaga (1976), Mizoguchi and Shimura (1980), and Wong and Lane (1983). Details of the algorithms are provided in the section "Clustering Methods" on page 2248.

For nonparametric clustering methods, a cluster is loosely defined as a region surrounding a local maximum of the probability density function (see the section "Significance Tests" on page 2251 for a more rigorous definition). Given a sufficiently large sample, nonparametric clustering methods are capable of detecting clusters of unequal size and dispersion and with highly irregular shapes. Nonparametric methods can also obtain good results for compact clusters of equal size and dispersion, but

they naturally require larger sample sizes for good recovery than clustering methods that are biased toward finding such “nice” clusters.

For coordinate data, nonparametric clustering methods are less sensitive to changes in scale of the variables or to affine transformations of the variables than are most other commonly used clustering methods. Nevertheless, it is necessary to consider questions of scaling and transformation, since variables with large variances tend to have more effect on the resulting clusters than those with small variances. If two or more variables are not measured in comparable units, some type of standardization or scaling is necessary; otherwise, the distances used by the procedure may be based on inappropriate apples-and-oranges computations. For variables with comparable units of measurement, standardization or scaling may still be desirable if the scale estimates of the variables are not related to their expected importance for defining clusters. If you want two variables to have equal importance in the analysis, they should have roughly equal scale estimates. If you want one variable to have more effect than another, the former should be scaled to have a greater scale estimate than the latter. The STD option in the PROC MODECLUS statement scales all variables to equal variance. However, the variance is not necessarily the most appropriate scale estimate for cluster analysis. In particular, outliers should be removed before using PROC MODECLUS with the STD option. A variety of scale estimators including robust estimators are provided in the STDIZE procedure (for detailed information, see Chapter 59, “The STDIZE Procedure”). Additionally, the ACECLUS procedure provides another way to transform the variables to try to improve the separation of clusters.

Since clusters are defined in terms of local maxima of the probability density function, nonlinear transformations of the data may change the number of population clusters. The variables should be transformed so that equal differences are of equal practical importance. An interval scale of measurement is required. Ordinal or ranked data are generally inappropriate, since monotone transformations can produce any arbitrary number of modes.

Unlike the methods in the CLUSTER procedure, the methods in the MODECLUS procedure are not inherently hierarchical. However, PROC MODECLUS can do approximate nonparametric significance tests for the number of clusters by obtaining an approximate  $p$ -value for each cluster, and it can hierarchically join nonsignificant clusters.

Another important difference between the MODECLUS procedure and many other clustering methods is that you do not tell PROC MODECLUS how many clusters you want. Instead, you specify a *smoothing parameter* (see the section “Density Estimation” on page 2245) and, optionally, a significance level, and PROC MODECLUS determines the number of clusters. You can specify a list of smoothing parameters, and PROC MODECLUS performs a separate cluster analysis for each value in the list.

## Getting Started

This section illustrates how PROC MODECLUS can be used to examine the clusters of data in the following artificial data set.

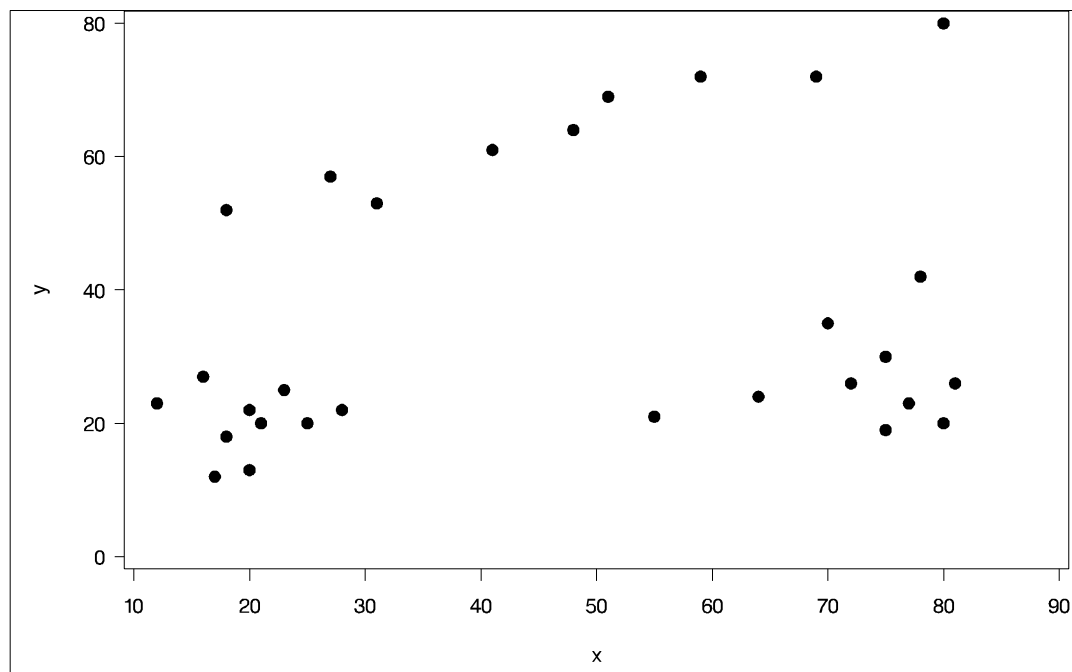
```
data example;
  input x y @@;
  datalines;
18 18 20 22 21 20 12 23 17 12 23 25 25 20 16 27
20 13 28 22 80 20 75 19 77 23 81 26 55 21 64 24
72 26 70 35 75 30 78 42 18 52 27 57 41 61 48 64
59 72 69 72 80 80 31 53 51 69 72 81
;
```

It is a good practice to plot the data to check for obvious clusters or pathologies prior to the analysis. The interactive graphics of the SAS/INSIGHT product are effective for visualizing clusters. In this example, with only two variables and a small sample size, the GPLOT procedure is adequate. The following statements produce Figure 42.1:

```
axis1 label=(angle=90 rotate=0) minor=none
      order=(0 to 80 by 20);
axis2 minor=none;
proc gplot;
  plot y*x /frame cframe=ligr vaxis=axis1 haxis=axis2;
run;
```

The plot suggests three clusters. Of these clusters, the one in the lower left corner is the most compact, while the lower right cluster is more dispersed.

The upper cluster is elongated and would be difficult for most clustering algorithms to identify as a single cluster. The plot also suggests that a Euclidean distance of 10 or 20 is a good initial guess for the neighborhood size in density estimation and clustering.



**Figure 42.1.** Scatter Plot of Data

To obtain a cluster analysis, you must specify the METHOD= option; for most purposes, METHOD=1 is recommended. The cluster analysis can be performed with a list of radii (R=10 15 35), as illustrated in the following PROC MODECLUS step. An output data set containing the cluster membership is created with the OUT= option and then used by PROC GPLOT to display the membership. The following statements produce Figure 42.2 through Figure 42.5:

```
proc modeclus data=example method=1 r=10 15 35 out=out;
run;
```

For each cluster solution, PROC MODECLUS produces a table of cluster statistics including the cluster number, the number of observations in the cluster, the maximum estimated density within the cluster, the number of observations in the cluster having a neighbor that belongs to a different cluster, and the estimated saddle density of the cluster. The results are displayed in Figure 42.2, Figure 42.3, and Figure 42.4 for three different radii. A smaller radius (R=10) yields a larger number of clusters (6), as displayed in Figure 42.1; a larger radius (R=35) includes all observations in a single cluster, as displayed in Figure 42.5. Note that all clusters in these three figures are “isolated” since their corresponding boundary frequencies are all 0s. Therefore, all the estimated saddle densities are missing.

The MODECLUS Procedure				
R=10 METHOD=1				
Cluster Statistics				
Cluster	Frequency	Maximum	Boundary	Estimated
		Estimated		Saddle
		Density	Frequency	Density
-----				
1	10	0.00106103	0	.
2	9	0.00084883	0	.
3	7	0.00031831	0	.
4	2	0.00021221	0	.
5	1	0.0001061	0	.
6	1	0.0001061	0	.

Figure 42.2. Results from PROC MODECLUS for METHOD=1 and R=10

The MODECLUS Procedure				
R=15 METHOD=1				
Cluster Statistics				
Cluster	Frequency	Maximum	Boundary	Estimated
		Estimated		Saddle
		Density	Frequency	Density
-----				
1	10	0.00047157	0	.
2	10	0.00042441	0	.
3	10	0.00023579	0	.

Figure 42.3. Results from PROC MODECLUS for METHOD=1 and R=15

The MODECLUS Procedure				
R=35 METHOD=1				
Cluster Statistics				
Cluster	Frequency	Maximum	Boundary	Estimated
		Estimated		Saddle
		Density	Frequency	Density
-----				
1	30	0.00012126	0	.

Figure 42.4. Results from PROC MODECLUS for METHOD=1 and R=35

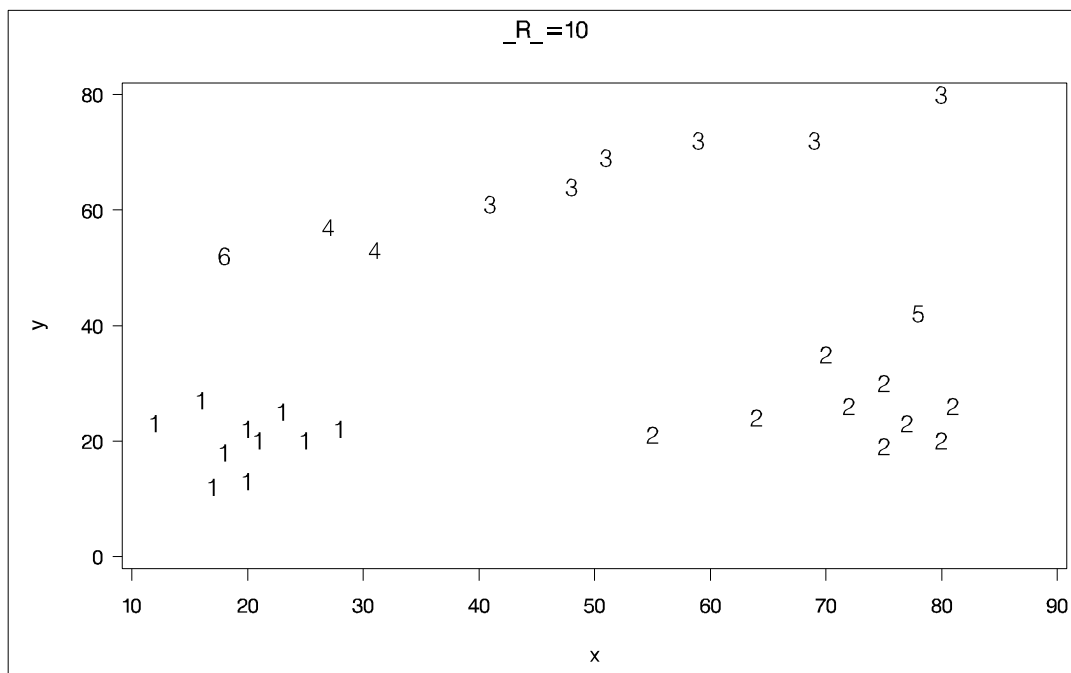
A table summarizing each cluster solution is then produced, as displayed in Figure 42.5.

The MODECLUS Procedure		
Cluster Summary		
R	Number of Clusters	Frequency of Unclassified Objects
-----		
10	6	0
15	3	0
35	1	0

Figure 42.5. Summary Table

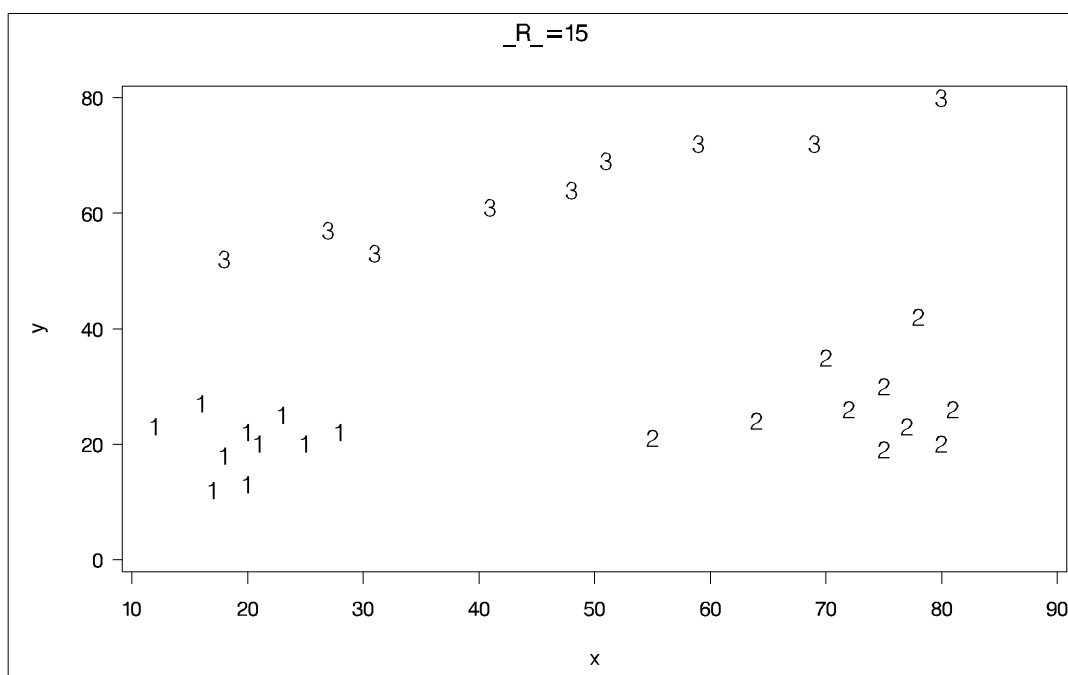
The OUT= data set contains a complete copy of the input data set for each cluster solution. Using a BY statement in the following PROC GPLOT step, you can examine the differences in cluster memberships for each radius. The following statements produce Figure 42.6 through Figure 42.8:

```
symbol1 v='1' font=swiss c=white; symbol2 v='2' font=swiss c=yellow;
symbol3 v='3' font=swiss c=cyan; symbol4 v='4' font=swiss c=green;
symbol5 v='5' font=swiss c=orange; symbol6 v='6' font=swiss c=blue;
symbol7 v='7' font=swiss c=black;
proc gplot data=out;
    plot y*x=cluster /frame cframe=ligr nolegend vaxis=axis1
        haxis=axis2;
by _r_;
run;
```

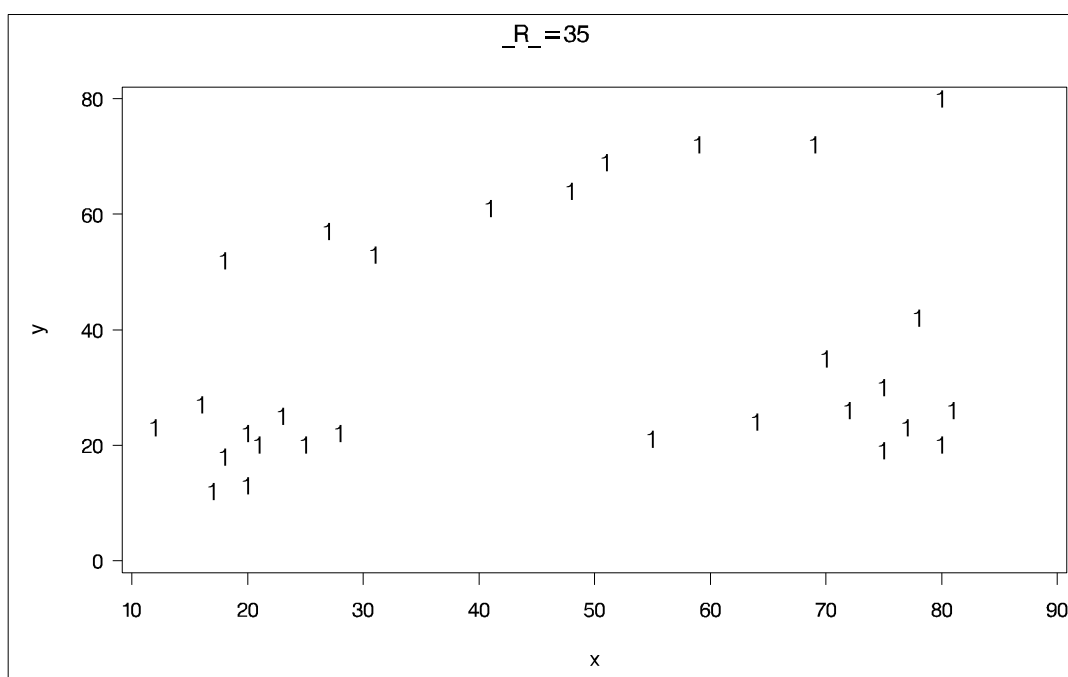


**Figure 42.6.** Scatter Plots of Cluster Memberships with `_R_=10`





**Figure 42.7.** Scatter Plots of Cluster Memberships with `_R_=15`



**Figure 42.8.** Scatter Plots of Cluster Memberships with `_R_=35`

## Syntax

The following statements invoke the MODECLUS procedure:

```
PROC MODECLUS < options > ;
  BY variables ;
  FREQ | FREQUENCY variable ;
  ID variable ;
  VAR | VARIABLES variables ;
```

The PROC MODECLUS statement is required. All other statements are optional.

## PROC MODECLUS Statement

```
PROC MODECLUS < options > ;
```

The PROC MODECLUS statement invokes the procedure.

Options available in the PROC MODECLUS statement are classified by function in Table 42.1. The corresponding default value for each option, if applicable, is also listed in this table.

**Table 42.1.** Functional Summary

Description	Option	Default Value
<b>Data Sets</b>		
specify input data set name	DATA=	_LAST_
specify output data set name for observations	OUT=	
specify output data set name for clusters	OUTC=	
specify output data set name for cluster solutions	OUTS=	
<b>Variables in Output Data Sets</b>		
specify variable in the OUT= and OUTCLUS= data sets identifying clusters	CLUSTER=	CLUSTER
specify variable in the OUT= data set containing density estimates	DENSITY=	DENSITY
specify length of variables in the output data sets	OUTLENGTH=	8
<b>Results and Data Processing before Clustering *</b>		
request simple statistics	SIMPLE	
standardize the variables to mean 0 and standard deviation 1	STANDARD	
<b>Smoothing Parameters</b>		

\*for coordinate data only

Table 42.1. (continued)

Description	Option	Default Value
specify number of neighbors to use for $k$ th-nearest-neighbor density estimation	DK=	
specify number of neighbors to use for clustering	CK=	
specify number of neighbors to use for $k$ th-nearest-neighbor density estimation and clustering	K=	
specify radius of the sphere of support for uniform-kernel density estimation	DR=	
specify radius of the neighborhood for clustering	CR=	
specify radius of the sphere of support for uniform-kernel density estimation and the neighborhood clustering	R=	
<b>Density Estimation Options</b>		
specify number of times the density estimates are to be cascaded	CASCADE=	0
compute the likelihood cross-validation criterion	CROSS or CROSSLIST	
specify dimensionality to be used when computing density estimates	DIMENSION=	nvar* or 1 <sup>†</sup>
use arithmetic means for cascading density estimates	AM	
use harmonic means for cascading density estimates	HM	
use sums for cascading density estimates	SUM	
<b>Clustering Methods Options</b>		
dissolve clusters with $n$ or fewer members	DOCK	
stop the analysis after obtaining a solution with either no cluster or a single cluster	EARLY	
request that nonsignificant clusters be hierarchically joined.	JOIN(=)	
specify maximum number of clusters to be obtained with METHOD=6	MAXCLUSTERS=	no limit
specify clustering method to use	METHOD=	
specify minimum members for either cluster to be designated a modal cluster when two clusters are joined using METHOD=5	MODE=	the value of K <sup>‡</sup> or 2 <sup>§</sup>
specify power of the density used with METHOD=6	POWER=	2

\*number of variables, for coordinate data

†for distance data

‡when K= is specified

§when K= is not specified

Table 42.1. (continued)

Description	Option	Default Value
specify approximate significance tests for the number of clusters	TEST	
specify assignment threshold used with METHOD=6	THRESHOLD=	0.5
<b>Miscellaneous Options</b>		
produce all optional output	ALL	
display the density and cluster membership of observations with neighbors belonging to a different cluster	BOUNDARY	
retain the neighbor lists for each observation in memory	CORE	
display the estimated cross-validated log density of each observation	CROSSLIST	
display the estimated density and cluster membership of each observation	LIST	
display estimates of local dimensionality and write them to the OUT=data set	LOCAL	
display the neighbors of each observation	NEIGHBOR	
suppress the display of the output	NOPRINT	
suppress the display of the summary of the number of clusters, number of unassigned observations, and maximum <i>p</i> -value for each analysis	NOSUMMARY	
suppress the display of statistics for each cluster	SHORT	
trace the cluster assignments for the METHOD=6 algorithm	TRACE	

You must specify at least one of the following options for smoothing parameters for density estimation: DK=, K=, DR=, or R=. To obtain a cluster analysis, you must specify the METHOD= option and at least one of the following smoothing parameters for clustering: CK=, K=, CR=, or R=. If you want significance tests for the number of clusters, you must specify either the DR= or R= option. See the section “Density Estimation” on page 2245 for a discussion of smoothing parameters.

You can specify lists of values for the DK=, CK=, K=, DR=, CR=, and R= options. Numbers in the lists can be separated by blanks or commas. You can include in the lists one or more items of the form *start TO stop BY increment*. Each list can contain either one value or the same number of values as in every other list that contains more than one value. If a list has only one value, that value is used in combination with all the values in longer lists. If two or more lists have more than one value, then one analysis is done using the first value in each list, another analysis is done using the second value in each list, and so on.

You can specify the following options in the PROC MODECLUS statement.

**ALL**

produces all optional output.

**AM**

specifies arithmetic means for cascading density estimates. See the description of the CASCADE= option.

**BOUNDARY**

displays the density and cluster membership of observations with neighbors belonging to a different cluster.

**CASCADE=*n***

**CASC=*n***

specifies the number of times the density estimates are to be cascaded (see the section “Density Estimation” on page 2245). The default value 0 performs no cascading.

You can specify a list of values for the CASCADE= option. Each value in the list is combined with each combination of smoothing parameters to produce a separate analysis.

**CK=*n***

specifies the number of neighbors to use for clustering. The number of neighbors should be at least 2 but less than the number of observations. See the section “Density Estimation” on page 2245.

**CLUSTER=*name***

provides a name for the variable in the OUT= and OUTCLUS= data sets identifying clusters. The default name is CLUSTER.

**CORE**

keeps the neighbor lists for each observation in the computer memory to make small problems run faster.

**CR=*n***

specifies the radius of the neighborhood for clustering. See the section “Density Estimation” on page 2245.

**CROSS**

computes the likelihood cross-validation criterion (Silverman 1986, pp. 52–55). This option appears to be of limited usefulness (see the section “Density Estimation” on page 2245).

**CROSSLIST**

displays the cross-validated log density of each observation.

**DATA=*SAS-data-set***

specifies the input data set containing observations to be clustered. If you omit the DATA= option, the most recently created SAS data set is used.

If the data set is TYPE=DISTANCE, the data are interpreted as a distance matrix. The number of variables must equal the number of observations in the data set or in each BY group. The distances are assumed to be Euclidean, but the procedure

accepts other types of distances or dissimilarities. Unlike the CLUSTER procedure, PROC MODECLUS uses the entire distance matrix, not just the lower triangle; the distances are not required to be symmetric. The neighbors of a given observation are determined solely from the distances in that observation. Missing values are considered infinite. Various distance measures can be computed from coordinate data using the %DISTANCE macro in the SAS/STAT sample library.

If the data set is not TYPE=DISTANCE, the data are interpreted as coordinates in a Euclidean space, and Euclidean distances are computed. The variables can be discrete or continuous and should be at the interval level of measurement.

**DENSITY=name**

provides a name for the variable in the OUT= data set containing density estimates. The default name is DENSITY.

**DIMENSION=n**

**DIM=n**

specifies the dimensionality to be used when computing density estimates. The default is the number of variables if the data are coordinates; the default is 1 if the data are distances.

**DK=n**

specifies the number of neighbors to use for  $k$ th-nearest-neighbor density estimation. The number of neighbors should be at least 2 but less than the number of observations. See the section “Density Estimation” on page 2245.

**DOCK=n**

dissolves clusters with  $n$  or fewer members by making the members unassigned.

**DR=n**

specifies the radius of the sphere of support for uniform-kernel density estimation. See the section “Density Estimation” on page 2245.

**EARLY**

stops the cluster analysis after obtaining either a solution with no cluster or a solution with one cluster to which all observations are assigned. The smoothing parameters should be specified in increasing order. This can reduce the computer time required for the analysis but may occasionally miss some multiple-cluster solutions.

**HM**

uses harmonic means for cascading density estimates. See the description of the CASCADE= option.

**JOIN<=p>**

requests that nonsignificant clusters be hierarchically joined. The JOIN option implies the TEST option. After each solution is obtained, the cluster with the largest approximate  $p$ -value is either joined to a neighboring cluster or, if there is no neighboring cluster, dissolved by making all of its members unassigned. After two clusters are joined, an analysis of the remaining clusters is displayed.

If you do not specify a  $p$ -value with the JOIN= option, joining continues until only one cluster remains, and the results are written to the output data sets after each analysis. If you specify a  $p$ -value with the JOIN= option, joining continues until the greatest approximate  $p$ -value is less than the value given in the JOIN= option, and only if there is more than one cluster are the results for that analysis written to the output data sets.

Any value of  $p$  less than  $1\text{E}-8$  is set to  $1\text{E}-8$ .

**K= $n$**

specifies the number of neighbors to use for  $k$ th-nearest-neighbor density estimation and clustering. The number of neighbors should be at least 2 but less than the number of observations. Specifying K= $n$  is equivalent to specifying both DK= $n$  and CK= $n$ . See the section “Density Estimation” on page 2245.

**LIST**

displays the estimated density and cluster membership of each observation.

**LOCAL**

requests estimates of local dimensionality (Tukey and Tukey 1981, pp. 236–237).

**MAXCLUSTERS= $n$**

**MAXC= $n$**

specifies the maximum number of clusters to be obtained with the METHOD=6 option. By default, there is no fixed limit.

**METHOD= $n$**

**MET= $n$**

**M= $n$**

specifies what clustering method to use. Since these methods do not have widely recognized names, the methods are indicated by numbers from 0 to 6. The methods are described in the section “Clustering Methods” on page 2248. For most purposes, METHOD=1 is recommended, although METHOD=6 may occasionally produce better results in return for considerably greater computer time and space requirements. METHOD=1 is not good for discrete coordinate data with only a few equally spaced values. In this case, METHOD=6 or METHOD=3 works better. METHOD=4 or METHOD=5 is less desirable than other methods when there are ties, since a general characteristic of agglomerative hierarchical clustering methods is that the results are indeterminate in the presence of ties.

You must specify the METHOD= option to obtain a cluster analysis.

You can specify a list of values for the METHOD= option. Each value in the list is combined with each combination of smoothing and cascading parameters to produce a separate cluster analysis.

**MODE= $n$**

specifies that when two clusters are joined using the METHOD=5 option (no other methods are affected by the MODE= option), each must have at least  $n$  members for either cluster to be designated a modal cluster. In any case, each cluster must also have a maximum density greater than the fusion density for either cluster to be designated a modal cluster. If you specify the K= option, the default value of

the `MODE=` option is the same as the value of the `K=` option because the use of  $k$ th-nearest-neighbor density estimation limits the resolution that can be obtained for clusters with fewer than  $k$  members. If you do not specify the `K=` option, the default is `MODE=2`. If you specify `MODE=0`, the default value is used instead of 0. If you specify a `FREQ` statement, the `MODE=` value is compared to the number of observations in each cluster, not to the sum of the frequencies.

#### **NEIGHBOR**

displays the neighbors of each observation in a table called “Nearest Neighbor List.”

#### **NOPRINT**

suppresses the display of the output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 15, “Using the Output Delivery System.”

#### **NOSUMMARY**

suppresses the display of the summary of the number of clusters, number of unassigned observations, and maximum  $p$ -value for each analysis.

#### **OUT=SAS-data-set**

specifies the output data set containing the input data plus density estimates, cluster membership, and variables identifying the type of solution. There is an output observation corresponding to each input observation for each solution. Therefore, the `OUT=` data set can be very large.

#### **OUTCLUS=SAS-data-set**

#### **OUTC=SAS-data-set**

specifies the output data set containing an observation corresponding to each cluster in each solution. The variables identify the solution and contain statistics describing the clusters.

#### **OUTSUM=SAS-data-set**

#### **OUTS=SAS-data-set**

specifies the output data set containing an observation corresponding to each cluster solution giving the number of clusters and the number of unclassified observations for that solution.

#### **OUTLENGTH= $n$**

#### **OUTL= $n$**

specifies the length of those output variables that are not copied from the input data set but are created by PROC MODECLUS.

The `OUTLENGTH=` option applies only to the following variables that appear in all of the output data sets: `_K_`, `_DK_`, `_CK_`, `_R_`, `_DR_`, `_CR_`, `_CASCAD_`, `_METHOD_`, `_NJOIN_`, and `_LOCAL_`.

The minimum value is 2 or 3, depending on the operating system. The maximum value is 8. The default value is 8.



**POWER=*n*****POW=*n***

specifies the power of the density used with the METHOD=6 option. The default value is 2.

**R=*n***

specifies the radius of the sphere of support for uniform-kernel density estimation and the neighborhood for clustering. Specifying R=*n* is equivalent to specifying both DR=*n* and CR=*n*. See the section “Density Estimation” on page 2245.

**SHORT**

suppresses the display of statistics for each cluster.

**SIMPLE****S**

displays means, standard deviations, skewness, kurtosis, and a coefficient of bimodality. The SIMPLE option applies only to coordinate data.

**STANDARD****STD**

standardizes the variables to mean 0 and standard deviation 1. The STANDARD option applies only to coordinate data.

**SUM**

uses sums for cascading density estimates. See the description of the CASCADE= option on page 2239.

**TEST**

performs approximate significance tests for the number of clusters. The R= or DR= option must also be specified with a nonzero value to obtain significance tests.

The significance tests performed by PROC MODECLUS are valid only for simple random samples, and they require at least 20 observations per cluster to have enough power to be of any use. See the section “Significance Tests” on page 2251.

**THRESHOLD=*n*****THR=*n***

specifies the assignment threshold used with the METHOD=6 option. The default is 0.5.

**TRACE**

traces the cluster assignments for the METHOD=6 algorithm.

---

## BY Statement

**BY** *variables* ;

You can specify a BY statement with PROC MODECLUS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the MODECLUS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

---

## FREQ Statement

**FREQ** | **FREQUENCY** *variable* ;

If one variable in the input data set represents the frequency of occurrence for other values in the observation, specify the variable's name in a FREQ statement. PROC MODECLUS then treats the data set as if each observation appeared  $n$  times, where  $n$  is the value of the FREQ variable for the observation. Nonintegral values of the FREQ variable are truncated to the largest integer less than the FREQ value.

---

## ID Statement

**ID** *variable* ;

The values of the ID variable identify observations in the displayed results and in the OUT= data set. If you omit the ID statement, each observation is identified by its observation number, and a variable called \_OBS\_ is written to the OUT= data set containing the original observation numbers.

---

## VAR Statement

**VAR** | **VARIABLES** *variables* ;

The VAR statement specifies numeric variables to be used in the cluster analysis. If you omit the VAR statement, all numeric variables not specified in other statements are used.

---

## Details

---

### Density Estimation

Refer to Silverman (1986) or Scott (1992) for an introduction to nonparametric density estimation.

PROC MODECLUS uses (hyper)spherical uniform kernels of fixed or variable radius. The density estimate at a point is computed by dividing the number of observations within a sphere centered at the point by the product of the sample size and the volume of the sphere. The size of the sphere is determined by the smoothing parameters that you are required to specify.

For fixed-radius kernels, specify the radius as a Euclidean distance with either the DR= or R= option. For variable-radius kernels, specify the number of neighbors desired within the sphere with either the DK= or K= option; the radius is then the smallest radius that contains at least the specified number of observations including the observation at which the density is being estimated. If you specify both the DR= or R= option and the DK= or K= option, the radius used is the maximum of the two indicated radii; this is useful for dealing with outliers.

It is convenient to refer to the sphere of support of the kernel at observation  $x_i$  as the *neighborhood* of  $x_i$ . The observations within the neighborhood of  $x_i$  are the *neighbors* of  $x_i$ . In some contexts,  $x_i$  is considered a neighbor of itself, but in other contexts it is not. The following notation is used in this chapter.

$x_i$	the $i$ th observation
$d(x,y)$	the distance between points $x$ and $y$
$n$	the total number of observations in the sample
$n_i$	the number of observations within the neighborhood of $x_i$ including $x_i$ itself
$n_i^-$	the number of observations within the neighborhood of $x_i$ not including $x_i$ itself
$N_i$	the set of indices of neighbors of $x_i$ including $i$
$N_i^-$	the set of indices of neighbors of $x_i$ not including $i$
$v_i$	the volume of the neighborhood of $x_i$
$\hat{f}_i$	the estimated density at $x_i$
$\hat{f}_i^-$	the cross-validated density estimate at $x_i$
$C_k$	the set of indices of observations assigned to cluster $k$
$v$	the number of variables or the dimensionality
$s_l$	standard deviation of the $l$ th variable

The estimated density at  $x_i$  is

$$\hat{f}_i = \frac{n_i}{nv_i}$$

that is, the number of neighbors of  $x_i$  divided by the product of the sample size and the volume of the neighborhood at  $x_i$ .

The density estimates provided by uniform kernels are not quite as good as those provided by some other types of kernels, but they are quite satisfactory for clustering. The significance tests for the number of clusters require the use of fixed-size uniform kernels.

There is no simple answer to the question of which smoothing parameter to use (Silverman 1986, pp. 43–61, 84–88, 98–99). It is usually necessary to try several different smoothing parameters. A reasonable first guess for the K= option is in the range of 0.1 to 1 times  $n^{4/(v+4)}$ , smaller values being suitable for higher dimensionalities. A reasonable first guess for the R= option in many coordinate data sets is given by

$$\left[ \frac{2^{v+2}(v+2)\Gamma(\frac{v}{2}+1)}{nv^2} \right]^{1/(v+4)} \sqrt{\sum_{l=1}^v s_l^2}$$

which can be computed in a DATA step using the GAMMA function for  $\Gamma$ . This formula is derived under the assumption that the data are sampled from a multivariate normal distribution and, therefore, tend to be too large (oversmooth) if the true distribution is multimodal. Robust estimates of the standard deviations may be preferable if there are outliers. If the data are distances, the factor  $\sum s_l^2$  can be replaced by an average (mean, trimmed mean, median, root-mean-square, and so on) distance divided by  $\sqrt{2}$ . To prevent outliers from appearing as separate clusters, you can also specify K=2 or CK=2 or, more generally, K= $m$  or CK= $m$ ,  $m \geq 2$ , which in most cases forces clusters to have at least  $m$  members.

If the variables all have unit variance (for example, if you specify the STD option), you can use Table 42.2 to obtain an initial guess for the R= option.

**Table 42.2.** Reasonable First Guess for R= for Standardized Data

Number of Obs	Number of Variables									
	1	2	3	4	5	6	7	8	9	10
20	1.01	1.36	1.77	2.23	2.73	3.25	3.81	4.38	4.98	5.60
35	0.91	1.24	1.64	2.08	2.56	3.08	3.62	4.18	4.77	5.38
50	0.84	1.17	1.56	1.99	2.46	2.97	3.50	4.06	4.64	5.24
75	0.78	1.09	1.47	1.89	2.35	2.85	3.38	3.93	4.50	5.09
100	0.73	1.04	1.41	1.82	2.28	2.77	3.29	3.83	4.40	4.99
150	0.68	0.97	1.33	1.73	2.18	2.66	3.17	3.71	4.27	4.85
200	0.64	0.93	1.28	1.67	2.11	2.58	3.09	3.62	4.17	4.75
350	0.57	0.85	1.18	1.56	1.98	2.44	2.93	3.45	4.00	4.56
500	0.53	0.80	1.12	1.49	1.91	2.36	2.84	3.35	3.89	4.45
750	0.49	0.74	1.06	1.42	1.82	2.26	2.74	3.24	3.77	4.32
1000	0.46	0.71	1.01	1.37	1.77	2.20	2.67	3.16	3.69	4.23
1500	0.43	0.66	0.96	1.30	1.69	2.11	2.57	3.06	3.57	4.11
2000	0.40	0.63	0.92	1.25	1.63	2.05	2.50	2.99	3.49	4.03

One data-based method for choosing the smoothing parameter is likelihood cross validation (Silverman 1986, pp. 52–55). The cross-validated density estimate at an observation is obtained by omitting the observation from the computations.

$$\hat{f}_i^- = \frac{n_i^-}{nv_i}$$

The (log) likelihood cross-validation criterion is then computed as

$$\sum_{i=1}^n \log \hat{f}_i^-$$

The suggested smoothing parameter is the one that maximizes this criterion. With fixed-radius kernels, likelihood cross validation oversmooths long-tailed distributions; for purposes of clustering, it tends to undersmooth short-tailed distributions. With  $k$ -nearest-neighbor density estimation, likelihood cross validation is useless because it almost always indicates  $k=2$ .

Cascaded density estimates are obtained by computing initial kernel density estimates and then, at each observation, taking the arithmetic mean, harmonic mean, or sum of the initial density estimates of the observations within the neighborhood. The cascaded density estimates can, in turn, be cascaded, and so on. Let  ${}_k\hat{f}_i$  be the density estimate at  $x_i$  cascaded  $k$  times. For all types of cascading,  ${}_0\hat{f}_i = \hat{f}_i$ . If the cascading is done by arithmetic means, then, for  $k \geq 0$ ,

$${}_{k+1}\hat{f}_i = \sum_{j \in N_i} {}_k\hat{f}_j / n_i$$

For harmonic means,

$${}_{k+1}\hat{f}_i = \left( \sum_{j \in N_i} {}_k\hat{f}_j^{-1} / n_i \right)^{-1}$$

and for sums,

$${}_{k+1}\hat{f}_i = \left( \sum_{j \in N_i} {}_k\hat{f}_j^{k+1} \right)^{\frac{1}{k+2}}$$

To avoid cluttering formulas, the symbol  $\hat{f}_i$  is used from now on to denote the density estimate at  $x_i$  whether cascaded or not, since the clustering methods and significance tests do not depend on the degree of cascading.

Cascading increases the smoothness of the estimates with less computation than would be required by increasing the smoothing parameters to yield a comparable degree of smoothness. For population densities with bounded support and discontinuities at the boundaries, cascading improves estimates near the boundaries. Cascaded estimates, especially using sums, may be more sensitive to the local covariance structure of the distribution than are the uncascaded kernel estimates. Cascading seems to be useful for detecting very nonspherical clusters. Cascading was suggested by Tukey and Tukey (1981, p. 237). Additional research into the properties of cascaded density estimates is needed.

---

## Clustering Methods

The number of clusters is a function of the smoothing parameters. The number of clusters tends to decrease as the smoothing parameters increase, but the relationship is not strictly monotonic. Generally, you should specify several different values of the smoothing parameters to see how the number of clusters varies.

The clustering methods used by PROC MODECLUS use spherical clustering neighborhoods of fixed or variable radius that are similar to the spherical kernels used for density estimation. For fixed-radius neighborhoods, specify the radius as a Euclidean distance with either the CR= or R= option. For variable-radius neighborhoods, specify the number of neighbors desired within the sphere with either the CK= or K= option; the radius is then the smallest radius that contains at least the specified number of observations including the observation for which the neighborhood is being determined. However, in the following descriptions of clustering methods, an observation is not considered to be one of its own neighbors. If you specify both the CR= or R= option and the CK= or K= option, the radius used is the maximum of the two indicated radii; this is useful for dealing with outliers. In this section, the symbols  $N_i$ ,  $N_i^-$ ,  $n_i$ , and  $n_i^-$  refer to clustering neighborhoods, not density estimation neighborhoods.

**METHOD=0**

Begin with each observation in a separate cluster. For each observation and each of its neighbors, join the cluster to which the observation belongs with the cluster to which the neighbor belongs. This method does not use density estimates. With a fixed clustering radius, the clusters are those obtained by cutting the single linkage tree at the specified radius (see Chapter 23, “The CLUSTER Procedure”).

**METHOD=1**

Begin with each observation in a separate cluster. For each observation, find the nearest neighbor with a greater estimated density. If such a neighbor exists, join the cluster to which the observation belongs with the cluster to which the specified neighbor belongs.

Next, consider each observation with density estimates equal to that of one or more neighbors but not less than the estimate at any neighbor. Join the cluster containing the observation with (1) each cluster containing a neighbor of the observation such that the maximum density estimate in the cluster equals the density estimate at the observation and (2) the cluster containing the nearest neighbor of the observation such that the maximum density estimate in the cluster exceeds the density estimate at the observation.

This method is similar to the classification or assignment stage of algorithms described by Gitman (1973) and Huizinga (1978).

**METHOD=2**

Begin with each observation in a separate cluster. For each observation, find the neighbor with the greatest estimated density exceeding the estimated density of the observation. If such a neighbor exists, join the cluster to which the observation belongs with the cluster to which the specified neighbor belongs.

Observations with density estimates equal to that of one or more neighbors but not less than the estimate at any neighbor are treated the same way as they are in METHOD=1.

This method is similar to the first stage of an algorithm proposed by Mizoguchi and Shimura (1980).

**METHOD=3**

Begin with each observation in a separate cluster. For each observation, find the neighbor with greater estimated density such that the slope of the line connecting the point on the estimated density surface at the observation with the point on the estimated density surface at the neighbor is a maximum. That is, for observation  $x_i$ , find a neighbor  $x_j$  such that  $(\hat{f}_j - \hat{f}_i)/d(x_j, x_i)$  is a maximum. If this slope is positive, join the cluster to which observation  $x_i$  belongs with the cluster to which the specified neighbor  $x_j$  belongs. This method was invented by Koontz, Narendra, and Fukunaga (1976).

Observations with density estimates equal to that of one or more neighbors but not less than the estimate at any neighbor are treated the same way as they are in METHOD=1. The algorithm suggested for this situation by Koontz, Narendra, and

Fukunaga (1976) may fail for flat areas in the estimated density that contain four or more observations.

#### **METHOD=4**

This method is equivalent to the first stage of two-stage density linkage (see Chapter 23, “The CLUSTER Procedure”) without the use of the MODE=option.

#### **METHOD=5**

This method is equivalent to the first stage of two-stage density linkage (see Chapter 23, “The CLUSTER Procedure”) with the use of the MODE=option.

#### **METHOD=6**

Begin with all observations unassigned.

**Step 1:** Form a list of seeds, each seed being a single observation such that the estimated density of the observation is not less than the estimated density of any of its neighbors. If you specify the MAXCLUSTERS= $n$  option, retain only the  $n$  seeds with the greatest estimated densities.

**Step 2:** Consider each seed in decreasing order of estimated density.

1. If the current seed has already been assigned, proceed to the next seed. Otherwise, form a new cluster consisting of the current seed.
2. Add to the cluster any unassigned seed that is a neighbor of a member of the cluster or that shares a neighbor with a member of the cluster; repeat until no unassigned seed satisfies these conditions.
3. Add to the cluster all neighbors of seeds that belong to the cluster.
4. Consider each unassigned observation. Compute the ratio of the sum of the  $p-1$  powers of the estimated density of the neighbors that belong to the current cluster to the sum of the  $p-1$  powers of the estimated density of all of its neighbors, where  $p$  is specified by the POWER= option and is 2 by default. Let  $x_i$  be the current observation, and let  $k$  be the index of the current cluster. Then this ratio is

$$r_{ik} = \frac{\sum_{j \in N_i \cap C_k} \hat{f}_j^{p-1}}{\sum_{j \in N_i} \hat{f}_j^{p-1}}$$

(The sum of the  $p-1$  powers of the estimated density of the neighbors of an observation is an estimate of the integral of the  $p$ th power of the density over the neighborhood.) If  $r_{ik}$  exceeds the maximum of 0.5 and the value of the THRESHOLD= option, add the observation  $x_i$  to the current cluster  $k$ . Repeat until no more observations can be added to the current cluster.

**Step 3:** (This step is performed only if the value of the THRESHOLD= option is less than 0.5.) Form a list of unassigned observations in decreasing order of estimated density. Repeat the following actions until the list is empty.



1. Remove the first observation from the list, for example, observation  $x_i$ .
2. For each cluster  $k$ , compute  $r_{ik}$ .
3. If the maximum over clusters of  $r_{ik}$  exceeds the value of the THRESHOLD= option, assign observation  $x_i$  to the corresponding cluster and insert all observations of which the current observation is a neighbor into the list, keeping the list in decreasing order of estimated density.

METHOD=6 is related to a method invented by Koontz and Fukunaga (1972a) and discussed by Koontz and Fukunaga (1972b).

---

## Significance Tests

Significance tests require that a fixed-radius kernel be specified for density estimation via the DR= or R= option. You can also specify the DK= or K= option, but only the fixed radius is used for the significance tests.

The purpose of the significance tests is as follows: given a simple random sample of objects from a population, obtain an estimate of the number of clusters in the population such that the probability in repeated sampling that the estimate exceeds the true number of clusters is not much greater than  $\alpha$ ,  $1\% \leq \alpha \leq 10\%$ . In other words, a sequence of null hypotheses of the form

$H_0^{(i)}$ : The number of population clusters is  $i$  or less

where  $i = 1, 2, \dots, n$ , is tested against the alternatives such as

$H_a^{(i)}$ : The number of population clusters exceeds  $i$

with a maximum experimentwise error rate of approximately  $\alpha$ . The tests protect you from overestimating the number of population clusters. It is impossible to protect against underestimating the number of population clusters without introducing much stronger assumptions than are used here, since the number of population clusters could conceivably exceed the sample size.

The method for conducting significance tests is as follows:

1. Estimate densities using fixed-radius uniform kernels.
2. Obtain preliminary clusters by a “valley-seeking” method. Other clustering methods could be used but would yield less power.
3. Compute an approximate  $p$ -value for each cluster by comparing the estimated maximum density in the cluster with the estimated maximum density on the cluster boundary.
4. Repeatedly join the least significant cluster with a neighboring cluster until all remaining clusters are significant.

5. Estimate the number of population clusters as the number of significant sample clusters.
6. The preceding steps can be repeated for any number of different radii, and the estimate of the number of population clusters can be taken to be the maximum number of significant sample clusters for any radius.

This method has the following useful features:

- No distributional assumptions are required.
- The choice of smoothing parameter is not critical since you can try any number of different values.
- The data can be coordinates or distances.
- Time and space requirements for the significance tests are no worse than those for obtaining the clusters.
- The power is high enough to be useful for practical purposes.

The method for computing the  $p$ -values is based on a series of plausible approximations. There are as yet no rigorous proofs that the method is infallible. Neither are there any asymptotic results. However, simulations for sample sizes ranging from 20 to 2000 indicate that the  $p$ -values are almost always conservative. The only case discovered so far in which the  $p$ -values are liberal is a uniform distribution in one dimension for which the simulated error rates exceed the nominal significance level only slightly for a limited range of sample sizes.

To make inferences regarding population clusters, it is first necessary to define what is meant by a cluster. For clustering methods using nonparametric density estimation, a cluster is usually loosely defined as a region surrounding a local maximum of the probability density function or a maximal connected set of local maxima. This definition may not be satisfactory for very rough densities with many local maxima. It is not applicable at all to discrete distributions for which the density does not exist. As another example in which this definition is not intuitively reasonable, consider a uniform distribution in two dimensions with support in the shape of a figure eight (including the interior). This density might be considered to contain two clusters even though it does not have two distinct modes.

These difficulties can be avoided by defining clusters in terms of the local maxima of a smoothed probability density or mass function. For example, define the neighborhood distribution function (NDF) with radius  $r$  at a point  $x$  as the probability that a randomly selected point will lie within a radius  $r$  of  $x$ , that is, the probability integral over a hypersphere of radius  $r$  centered at  $x$ :

$$s(x) = P(d(x, X) \leq r)$$

where  $X$  is the random variable being sampled,  $r$  is a user-specified radius, and  $d(x, y)$  is the distance between points  $x$  and  $y$ .

The NDF exists for all probability distributions. You can select the radius according to the degree of resolution required. The minimum-variance unbiased estimate of the NDF at a point  $x$  is proportional to the uniform-kernel density estimate with corresponding support.

You can define a *modal region* as a maximal connected set of local maxima of the NDF. A cluster is a connected set containing exactly one modal region. This definition seems to give intuitively reasonable results in most cases. An exception is a uniform density on the perimeter of a square. The NDF has four local maxima. There are eight local maxima along the perimeter, but running PROC MODECLUS with the R= option would yield four clusters since the two local maxima at each corner are separated by a distance equal to the radius. While this density does indeed have four distinctive features (the corners), it is not obvious that each corner should be considered a cluster.

The number of population clusters depends on the radius of the NDF. The significance tests in PROC MODECLUS protect against overestimating the number of clusters at any specified radius. It is often useful to look at the clustering results across a range of radii. A plot of the number of sample clusters as a function of the radius is a useful descriptive display, especially for high-dimensional data (Wong and Schaack 1982).

If a population has two clusters, it must have two modal regions. If there are two modal regions, there must be a “valley” between them. It seems intuitively desirable that the boundary between the two clusters should follow the bottom of this valley. All the clustering methods in PROC MODECLUS are designed to locate the estimated cluster boundaries in this way, although methods 1 and 6 seem to be much more successful at this than the others. Regardless of the precise location of the cluster boundary, it is clear that the maximum of the NDF along the boundary between two clusters must be strictly less than the value of the NDF in either modal region; otherwise, there would be only a single modal region; according to Hartigan and Hartigan (1985), there must be a “dip” between the two modes. PROC MODECLUS assesses the significance of a sample cluster by comparing the NDF in the modal region with the maximum of the NDF along the cluster boundary. If the NDF has second-order derivatives in the region of interest and if the boundary between the two clusters is indeed at the bottom of the valley, then the maximum value of the NDF along the boundary occurs at a saddle point. Hence, this test is called a *saddle test*. This term is intended to describe any test for clusters that compares modal densities with saddle densities, not just the test currently implemented in the MODECLUS procedure.

The obvious estimate of the maximum NDF in a sample cluster is the maximum estimated NDF at an observation in the cluster. Let  $m(k)$  be the index of the observation for which the maximum is attained in cluster  $k$ .

Estimating the maximum NDF on the cluster boundary is more complicated. One approach is to take the maximum NDF estimate at an observation in the cluster that has a neighbor belonging to another cluster. This method yields excessively large estimates when the neighborhood is large. Another approach is to try to choose an object closer to the boundary by taking the observation with the maximum sum of estimated densities of neighbors belonging to a different cluster. After some experimentation, it is found that a combination of these two methods works well. Let  $B_k$

be the set of indices of observations in cluster  $k$  that have neighbors belonging to a different cluster, and compute

$$\max_{i \in B_k} \left( 0.2 \hat{f}_i n_i + \sum_{j \in N_i - C_k} \hat{f}_j \right)$$

Let  $s(k)$  be the index of the observation for which the maximum is attained.

Using the notation  $\#(S)$  for the cardinality of set  $S$ , let

$$\begin{aligned} n_{ij}^- &= \#(N_i^- \cap N_j^-) \\ c_m(k) &= n_{m(k)}^- - n_{m(k)s(k)}^- \\ c_s(k) &= n_{s(k)}^- - n_{m(k)s(k)}^- \text{ if } B_k \neq \emptyset, \\ &= 0 \text{ otherwise} \\ q_k &= 1/2 \text{ if } B_k \neq \emptyset, \\ &= 2/3 \text{ otherwise} \\ z_k &= \frac{c_m(k) - q_k(c_m(k) + c_s(k)) - 1/2}{\sqrt{q_k(1 - q_k)(c_m(k) + c_s(k))}} \\ u &= \left\lceil (.2 + .05\sqrt{n}) \sum_{i:n_i > 1} \frac{1}{n_i + 1} \right\rceil \end{aligned}$$

Let  $R(u)$  be a random variable distributed as the range of a random sample of  $u$  observations from a standard normal distribution. Then the approximate  $p$ -value  $p_k$  for cluster  $k$  is

$$p_k = Pr(z_k > R(u)/\sqrt{2})$$

If points  $m(k)$  and  $s(k)$  are fixed a priori,  $z_k$  would be the usual approximately normal test statistic for comparing two binomial random variables. In fact,  $m(k)$  and  $s(k)$  are selected in such a way that  $c_m(k)$  tends to be large and  $c_s(k)$  tends to be small. For this reason, and because there may be a large number of clusters, each with its own  $z_k$  to be tested, each  $z_k$  is referred to the distribution of  $R(u)$  instead of a standard normal distribution. If the tests are conducted for only one radius and if  $u$  is chosen equal to  $n$ , then the  $p$ -values are very conservative because (1) you are not making all possible pairwise comparisons of observations in the sample and (2)  $n_i^-$  and  $n_j^-$  are positively correlated if the neighborhoods overlap. In the formula for  $u$ , the summation overcorrects somewhat for the conservativeness due to correlated  $n_i^-$ 's. The factor  $.2 + .05\sqrt{n}$  is empirically estimated from simulation results to adjust for the use of more than one radius.

If the JOIN option is specified, the least significant cluster (the cluster with the smallest  $z_k$ ) is either dissolved or joined with a neighboring cluster. If no members of

the cluster have neighbors belonging to a different cluster, all members of the cluster are unassigned. Otherwise, the cluster is joined to the neighboring cluster such that the sum of density estimates of neighbors of the estimated saddle point belonging to it is a maximum. Joining clusters increases the power of the saddle test. For example, consider a population with two well-separated clusters. Suppose that, for a certain radius, each population cluster is divided into two sample clusters. None of the four sample clusters is likely to be significant, but after the two sample clusters corresponding to each population cluster are joined, the remaining two clusters may be highly significant.

The saddle test implemented in PROC MODECLUS has been evaluated by simulation from known distributions. Some results are given in the following three tables. In Table 42.3, samples of 20 to 2000 observations are generated from a one-dimensional uniform distribution. For sample sizes of 1000 or less, 2000 samples are generated and analyzed by PROC MODECLUS. For a sample size of 2000, only 1000 samples are generated. The analysis is done with at least 20 different values of the R= option spread across the range of radii most likely to yield significant results. The six central columns of the table give the observed error rates at the nominal error rates ( $\alpha$ ) at the head of each column. The standard errors of the observed error rates are given at the bottom of the table. The observed error rates are conservative for  $\alpha \leq 5\%$ , but they increase with  $\alpha$  and become slightly liberal for sample sizes in the middle of the range tested.

**Table 42.3.** Observed Error Rates (%) for Uniform Distribution

Sample Size	Nominal Type 1 Error Rate						Number of Simulations
	1	2	5	10	15	20	
20	0.00	0.00	0.00	0.60	11.65	27.05	2000
50	0.35	0.70	4.50	10.95	20.55	29.80	2000
100	0.35	0.85	3.90	11.05	18.95	28.05	2000
200	0.30	1.35	4.00	10.50	18.60	27.05	2000
500	0.45	1.05	4.35	9.80	16.55	23.55	2000
1000	0.70	1.30	4.65	9.55	15.45	19.95	2000
2000	0.40	1.10	3.00	7.40	11.50	16.70	1000
Standard Error	0.22	0.31	0.49	0.67	0.80	0.89	2000
	0.31	0.44	0.69	0.95	1.13	1.26	1000

All unimodal distributions other than the uniform that have been tested, including normal, Cauchy, and exponential distributions and uniform mixtures, have produced much more conservative results. Table 42.4 displays results from a unimodal mixture of two normal distributions with equal variances and equal sampling probabilities and with means separated by two standard deviations. Any greater separation would produce a bimodal distribution. The observed error rates are quite conservative.

**Table 42.4.** Observed Error Rates (%) for Normal Mixture with  $2\sigma$  Separation

Sample Size	Nominal Type 1 Error Rate						Number of Simulations
	1	2	5	10	15	20	
100	0.0	0.0	0.0	1.0	2.0	4.0	200
200	0.0	0.0	0.0	2.0	3.0	3.0	200
500	0.0	0.0	0.5	0.5	0.5	0.5	200

All distributions in two or more dimensions that have been tested yield extremely conservative results. For example, a uniform distribution on a circle yields observed error rates that are never more than one-tenth of the nominal error rates for sample sizes up to 1000. This conservatism is due to the fact that, as the dimensionality increases, more and more of the probability lies in the tails of the distribution (Silverman 1986, p. 92), and the saddle test used by PROC MODECLUS is more conservative for distributions with pronounced tails. This applies even to a uniform distribution on a hypersphere because, although the density has no tails, the NDF does.

Since the formulas for the significance tests do not involve the dimensionality, no problems are created when the data are linearly dependent. Simulations of data in nonlinear subspaces (the circumference of a circle or surface of a sphere) have also yielded conservative results.

Table 42.5 displays results in terms of power for identifying two clusters in samples from a bimodal mixture of two normal distributions with equal variances and equal sampling probabilities separated by four standard deviations. In this simulation, PROC MODECLUS never indicated more than two significant clusters.

**Table 42.5.** Power (%) for Normal Mixture with  $4\sigma$  Separation

Sample Size	Nominal Type 1 Error Rate						Number of Simulations
	1	2	5	10	15	20	
20	0.0	0.0	0.0	2.0	37.5	68.5	200
35	0.0	13.5	38.5	48.5	64.0	75.5	200
50	17.5	26.0	51.5	67.0	78.5	84.0	200
75	25.5	36.0	58.5	77.5	85.5	89.5	200
100	40.0	54.5	72.5	84.5	91.5	92.5	200
150	70.5	80.0	92.0	97.0	100.0	100.0	200
200	89.0	96.0	99.5	100.0	100.0	100.0	200

The saddle test is not as efficient as excess-mass tests for multimodality (Müller and Sawitzki 1991, Polonik 1993). However, there is not yet a general approximation for the distribution of excess-mass statistics to circumvent the need for simulations to do significance tests. Refer to Minnotte (1992) for a review of tests for multimodality.

---

## Computational Resources

The MODECLUS procedure stores coordinate data in memory if there is enough space. For distance data, only one observation at a time is in memory.

PROC MODECLUS constructs lists of the neighbors of each observation. The total space required is  $12 \sum n_i$  bytes, where  $n_i$  is based on the largest neighborhood required by any analysis. The lists are stored in a SAS utility data set unless you specify the CORE option. You may get an error message from the SAS System or from the operating system if there is not enough disk space for the utility data set. Clustering method 6 requires a second list that is always stored in memory.

For coordinate data, the time required to construct the neighbor lists is roughly proportional to  $v(\log n)(\sum n_i \log(\sum n_i/n))$ . For distance data, the time is roughly proportional to  $n^2 \log(\sum n_i/n)$ .

The time required for density estimation is proportional to  $\sum n_i$  and is usually small compared to the time required for constructing the neighbor lists.

Clustering methods 0 through 3 are quite efficient, requiring time proportional to  $\sum n_i$ . Methods 4 and 5 are slower, requiring time roughly proportional to  $(\sum n_i) \log(\sum n_i)$ . Method 6 can also be slow, but the time requirements depend very much on the data and the particular options specified. Methods 4, 5, and 6 also require more memory than the other methods.

The time required for significance tests is roughly proportional to  $g \sum n_i$ , where  $g$  is the number of clusters.

PROC MODECLUS can process data sets of several thousand observations if you specify reasonable smoothing parameters. Very small smoothing values produce many clusters, whereas very large values produce many neighbors; either case can require excessive time or space.

---

## Missing Values

If the data are coordinates, observations with missing values are excluded from the analysis.

If the data are distances, missing values are treated as infinite. The neighbors of each observation are determined solely by the distances in that observation. The distances are not required to be symmetric, and there is no check for symmetry; the neighbors of each observation are determined only from the distances in that observation. This treatment of missing values is quite different from that of the CLUSTER procedure, which ignores the upper triangle of the distance matrix.



---

## Output Data Sets

The OUT= data set contains one complete copy of the input data set for each cluster solution. There are additional variables identifying each solution and giving information about individual observations. Solutions with only one remaining cluster when JOIN=*p* is specified are omitted from the OUT= data set (see the description of the JOIN= option on page 2240). The OUT= data set can be extremely large, so it may be advisable to specify the DROP= data set option to exclude unnecessary variables.

The OUTCLUS= or OUTC= data set contains one observation for each cluster in each cluster solution. The variables identify the solution and provide statistics describing the cluster.

The OUTSUM= or OUTS= data set contains one observation for each cluster solution. The variables identify the solution and provide information about the solution as a whole.

The following variables can appear in all of the output data sets:

- `_K_`, which is the value of the K= option for the current solution. This variable appears only if you specify the K= option.
- `_DK_`, which is the value of the DK= option for the current solution. This variable appears only if you specify the DK= option.
- `_CK_`, which is the value of the CK= option for the current solution. This variable appears only if you specify the CK= option.
- `_R_`, which is the value of the R= option for the current solution. This variable appears only if you specify the R= option.
- `_DR_`, which is the value of the DR= option for the current solution. This variable appears only if you specify the DR= option.
- `_CR_`, which is the value of the CR= option for the current solution. This variable appears only if you specify the CR= option.
- `_CASCAD_`, which is the number of times the density estimates have been cascaded for the current solution. This variable appears only if you specify the CASCADE= option.
- `_METHOD_`, which is the value of the METHOD= option for the current solution. This variable appears only if you specify the METHOD= option.
- `_NJOIN_`, which is the number of clusters that are joined or dissolved in the current solution. This variable appears only if you specify the JOIN option.
- `_LOCAL_`, which is the local dimensionality estimate of the observation. This variable appears only if you specify the LOCAL option.



The OUT= data set contains the following variables:

- the variables from the input data set
- **\_OBS\_**, which is the observation number from the input data set. This variable appears only if you omit the ID statement.
- **DENSITY**, which is the estimated density at the observation. This variable can be renamed by the DENSITY= option.
- **CLUSTER**, which is the number of the cluster to which the observation is assigned. This variable can be renamed by the CLUSTER= option.

The OUTC= data set contains the following variables:

- the BY variables, if any
- **\_NCLUS\_**, which is the number of clusters in the solution
- **CLUSTER**, which is the number of the current cluster
- **\_FREQ\_**, which is the number of observations in the cluster
- **\_MODE\_**, which is the maximum estimated density in the cluster
- **\_BFREQ\_**, which is the number of observations in the cluster with neighbors belonging to a different cluster
- **\_SADDLE\_**, which is the estimated saddle density for the cluster
- **\_MC\_**, which is the number of observations within the fixed-radius density-estimation neighborhood of the modal observation. This variable appears only if you specify the TEST or JOIN option.
- **\_SC\_**, which is the number of observations within the fixed-radius density-estimation neighborhood of the saddle observation. This variable appears only if you specify the TEST or JOIN option.
- **\_OC\_**, which is the number of observations within the overlap of the two previous neighborhoods. This variable appears only if you specify the TEST or JOIN option.
- **\_Z\_**, which is the approximate  $z$  statistic for the cluster. This variable appears only if you specify the TEST or JOIN option.
- **\_P\_**, which is the approximate  $p$ -value for the cluster. This variable appears only if you specify the TEST or JOIN option.

The OUTS= data set contains the following variables:

- the BY variables, if any
- **\_NCLUS\_**, which is the number of clusters in the solution
- **\_UNCL\_**, which is the number of unclassified observations
- **\_CROSS\_**, which is the likelihood cross-validation criterion if you specify the CROSS or CROSSLIST option

---

## Displayed Output

If you specify the **SIMPLE** option and the data are coordinates, PROC MODECLUS displays the following simple descriptive statistics for each variable:

- the MEAN
- the standard deviation, STD DEV
- the SKEWNESS
- the KURTOSIS
- a coefficient of BIMODALITY (see Chapter 23, “The CLUSTER Procedure”)

If you specify the **NEIGHBOR** option, PROC MODECLUS displays a list of the neighbors of each observation. The table contains

- the observation number or ID value of the observation
- the observation number or ID value of each of its neighbors
- the distance to each neighbor

If you specify the **CROSSLIST** option, PROC MODECLUS produces a table of information regarding cross validation of the density estimates. Each table has a row for each observation. For each observation, the following are displayed:

- the observation number or ID value of the observation
- the radius of the neighborhood
- the number of neighbors
- the estimated log density
- the estimated cross-validated log density

If you specify the **LOCAL** option, PROC MODECLUS produces a table of information regarding estimates of local dimensionality. Each table has a row for each observation. For each observation, the following are displayed:

- the observation number or ID value of the observation
- the radius of the neighborhood
- the estimated local dimensionality

If you specify the LIST option, PROC MODECLUS produces a table listing the observations within each cluster. The table can include

- the cluster number
- the observation number or ID value of the observation
- the estimated density
- the sum of the density estimates of observations within the neighborhood that belong to the same cluster
- the sum of the density estimates of observations within the neighborhood that belong to a different cluster
- the sum of the density estimates of all the observations within the neighborhood
- the ratio of the sum of the density estimates for the same cluster to the sum of all the density estimates in the neighborhood

If you specify the LIST option and there are unassigned objects, PROC MODECLUS produces a table listing those observations. The table includes

- the observation number or ID value of the observation
- the estimated density
- the ratio of the sum of the density estimates for the same cluster to the sum of the density estimates in the neighborhood for all other clusters

If you specify the BOUNDARY option, PROC MODECLUS produces a table listing the observations in each cluster that have a neighbor belonging to a different cluster. The table includes

- the observation number or ID value of the observation
- the estimated density
- the cluster number
- the ratio of the sum of the density estimates for the same cluster to the sum of the density estimates in the neighborhood for all other clusters

If you do not specify the SHORT option, PROC MODECLUS produces a table of cluster statistics including

- the cluster number
- the cluster frequency (the number of observations in the cluster)
- the maximum estimated density within the cluster

- the number of observations in the cluster having a neighbor that belongs to a different cluster
- the estimated saddle density of the cluster

If you specify the TEST or JOIN option, the table of cluster statistics includes the following items pertaining to the saddle test:

- the number of observations within the fixed-radius density-estimation neighborhood of the modal observation
- the number of observations within the fixed-radius density-estimation neighborhood of the saddle observation
- the number of observations within the overlap of the two preceding neighborhoods
- the  $z$  statistic for comparing the preceding counts
- the approximate  $p$ -value

If you do not specify the NOSUMMARY option, PROC MODECLUS produces a table summarizing each cluster solution containing the following items:

- the smoothing parameters and cascade value
- the number of clusters
- the frequency of unclassified objects
- the likelihood cross-validation criterion if you specify the CROSS or CROSSLIST option

If you specify the JOIN option, the summary table also includes

- the number of clusters joined
- the maximum  $p$ -value of any cluster in the solution

If you specify the TRACE option, PROC MODECLUS produces a table for each cluster solution that lists each observation along with its cluster membership as it is reassigned from the “Old” cluster to the “New” cluster. This reassignment is described in **Step 1** through **Step 3** of the section “METHOD=6” on page 2250. Each table has a row for each observation. For each observation, the following are displayed:

- the observation number or ID value of the observation
- the estimated density
- the “Old” cluster membership. 0 represents an unassigned observation and -1 represents a seed.
- the “New” cluster membership

- “Ratio,” which is documented in the section “METHOD=6” on page 2250. The following character values can also be displayed:

“M”	means the observation is a mode
“S”	means the observation is a seed
“N”	means the neighbor of a mode or seed, for which the ratio is not computed

## ODS Table Names

PROC MODECLUS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 15, “Using the Output Delivery System.”

**Table 42.6.** ODS Tables Produced in PROC MODECLUS

ODS Table Name	Description	Statement	Option
BoundaryFreq	Boundary objects information	PROC	BOUNDARY (or ALL)
ClusterList	Cluster listing, cluster id, freq., density etc.	PROC	LIST (or ALL)
ClusterStats	Cluster statistics	PROC	default
	Cluster statistics, significance test statistics	PROC	TEST or JOIN (or ALL)
ClusterSummary	Cluster summary	PROC	default
	Cluster summary, crossvalidation criterion	PROC	CROSS or CROSSTEST (or ALL)
	Cluster summary, clusters joined information	PROC	JOIN (or ALL)
CrossList	Cross-validated log density	PROC	CROSSTEST
ListLocal	Local dimensionality estimates	PROC	LOCAL
Neighbor	Nearest neighbor list	PROC	NEIGHBOR (or ALL)
SimpleStatistics	Simple statistics	PROC	SIMPLE (or ALL)
Trace	Trace of clustering algorithm (METHOD= 6 only)	PROC	TRACE (or ALL) with METHOD= 6
UnassignObjects	Information on unassigned objects	PROC	LIST (or ALL)

## Examples

### Example 42.1. Cluster Analysis of Samples from Univariate Distributions

This example uses pseudo-random samples from a uniform distribution, an exponential distribution, and a bimodal mixture of two normal distributions. Results are presented in Output 42.1.1 through Output 42.1.3 as plots displaying both the true density and the estimated density, as well as cluster membership.

The following statements produce Output 42.1.1:

```
options noovp ps=28 ls=95;
title 'Modeclus Example with Univariate Distributions';
title2 'Uniform Distribution';

data uniform;
  drop n;
  true=1;
  do n=1 to 100;
    x=ranuni(123);
    output;
  end;

axis1 label=(angle=90 rotate=0) minor=none
      order=(0 to 3 by 0.5);
axis2 minor=none;
symbol9 v=none i=splines;

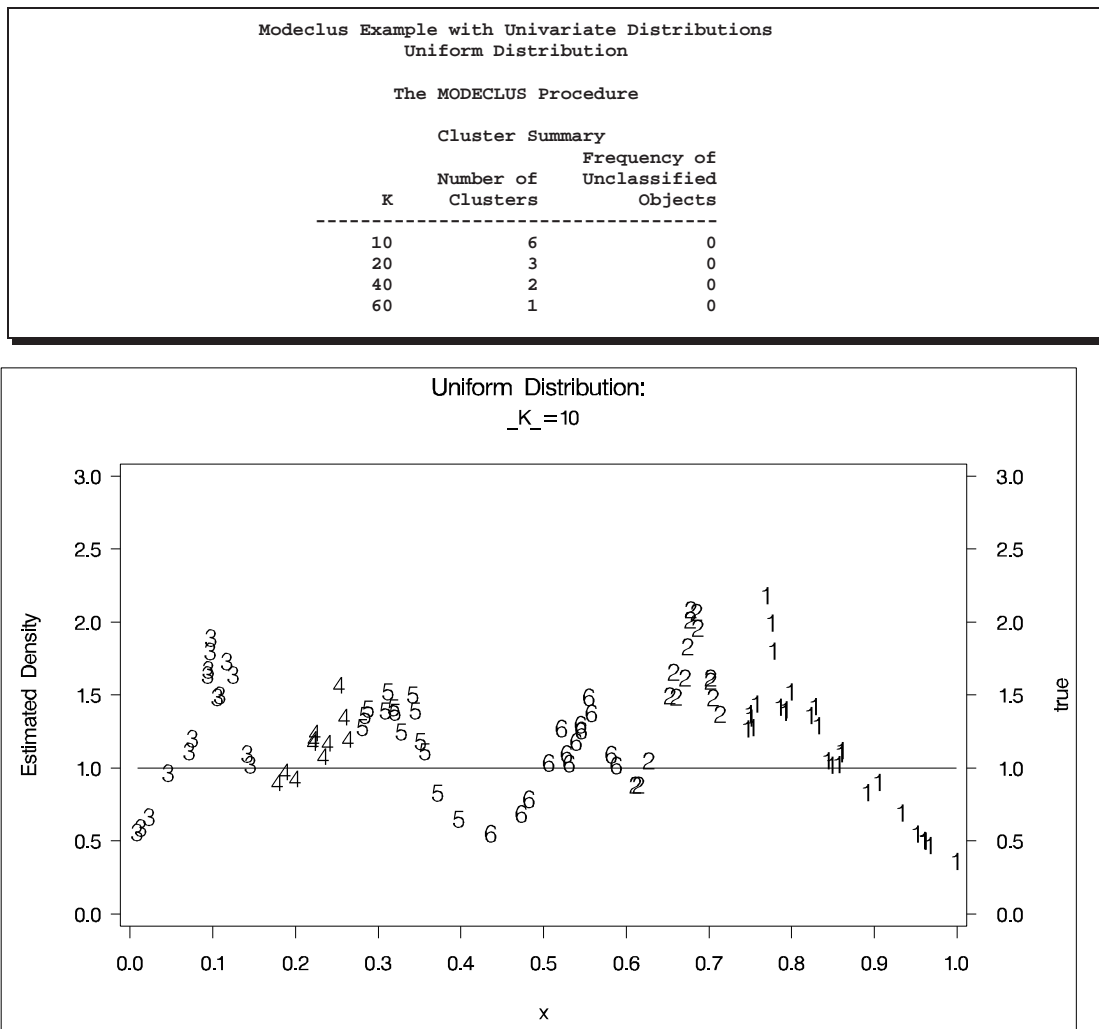
proc modeclus data=uniform m=1 k=10 20 40 60 out=out short;
  var x;

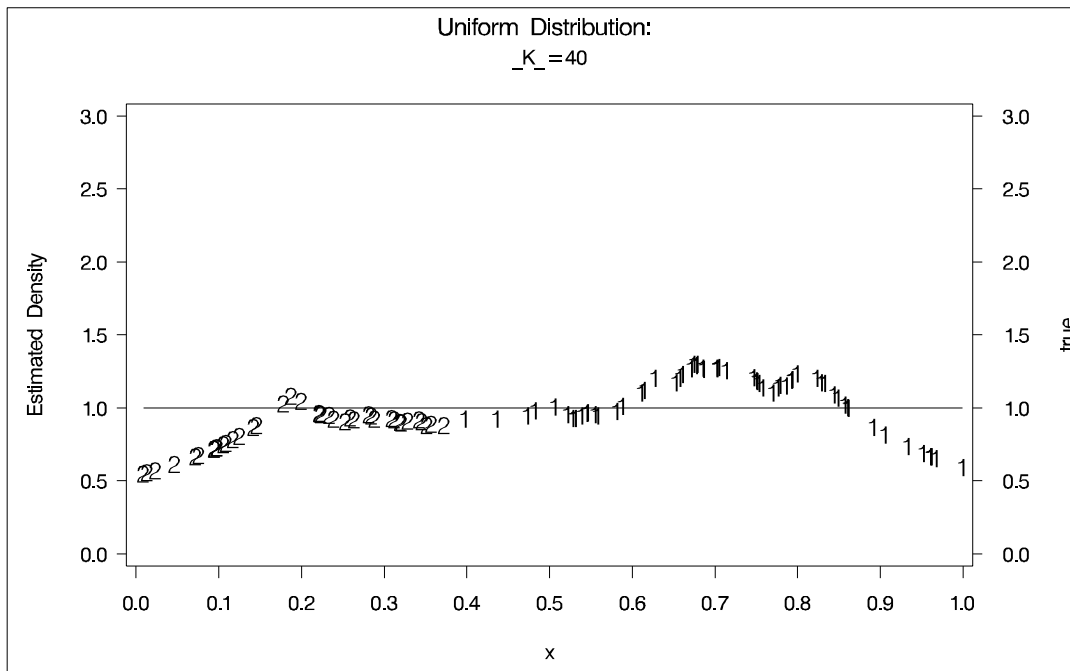
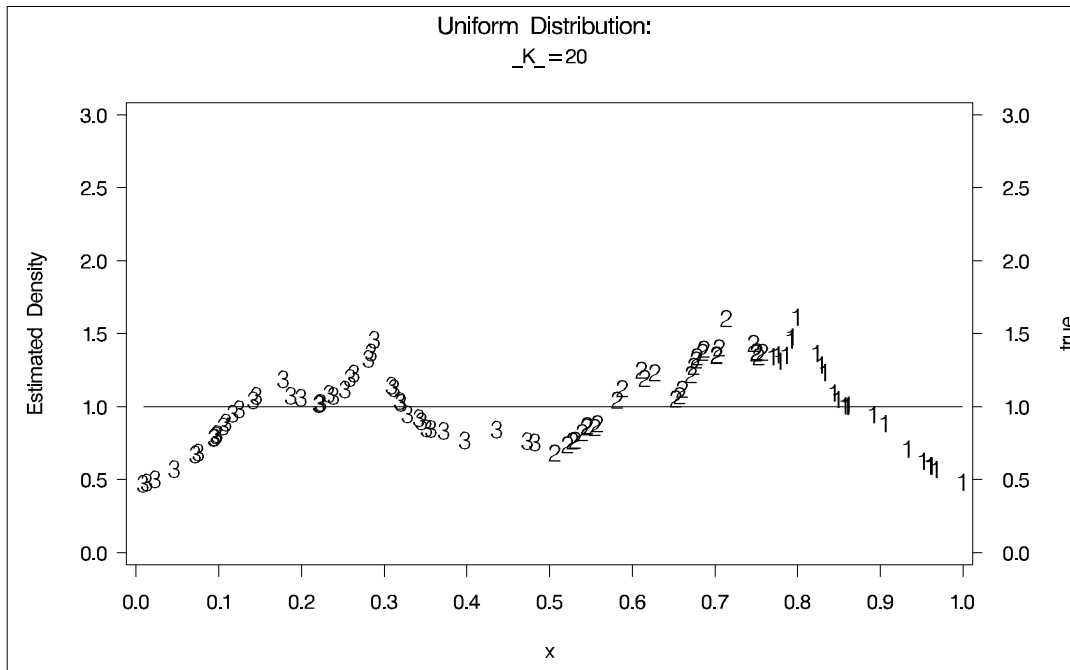
proc gplot data=out;
  plot density*x=cluster /frame cframe=ligr
                        vzero nolegend
                        vaxis=axis1 haxis=axis2;
  plot2 true*x=9/vaxis=axis1 haxis=axis2;
  by _K_;
run;
```

```
proc modeclus data=uniform m=1 r=.05 .10 .20 .30
              out=out short;
  var x;

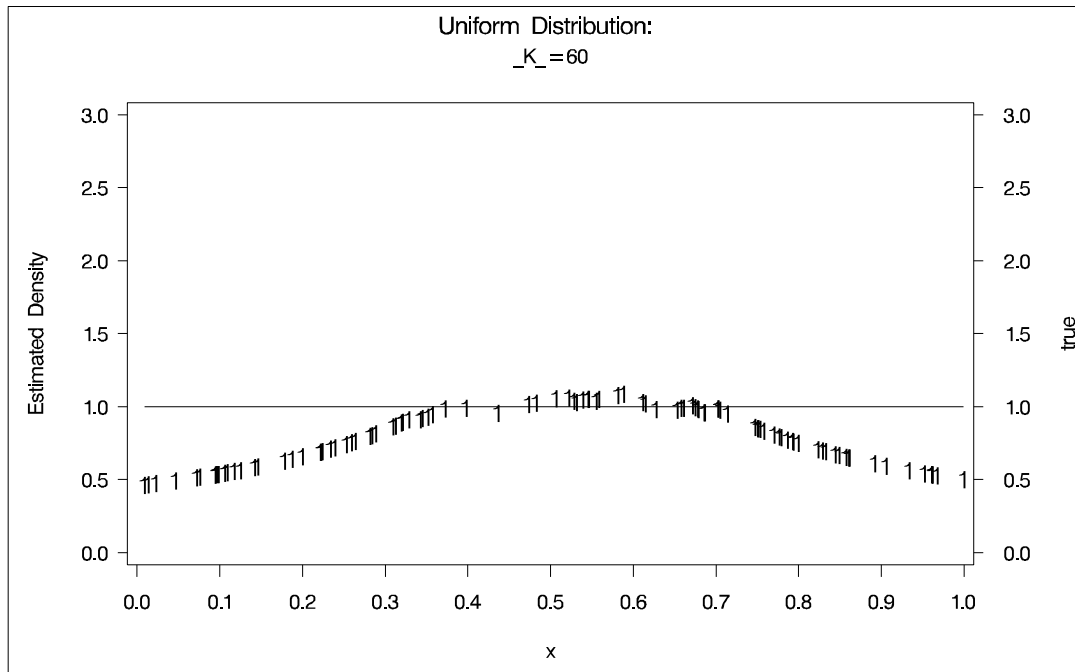
  axis1 label=(angle=90 rotate=0)
        minor=none order=(0 to 2 by 0.5);
proc gplot data=out;
  plot density*x=cluster /frame cframe=ligr
                        vzero nolegend
                        vaxis=axis1 haxis=axis2;
  plot2 true*x=9/vaxis=axis1 haxis=axis2;
  by _R_;
run;
```

**Output 42.1.1.** Cluster Analysis of Sample from a Uniform Distribution





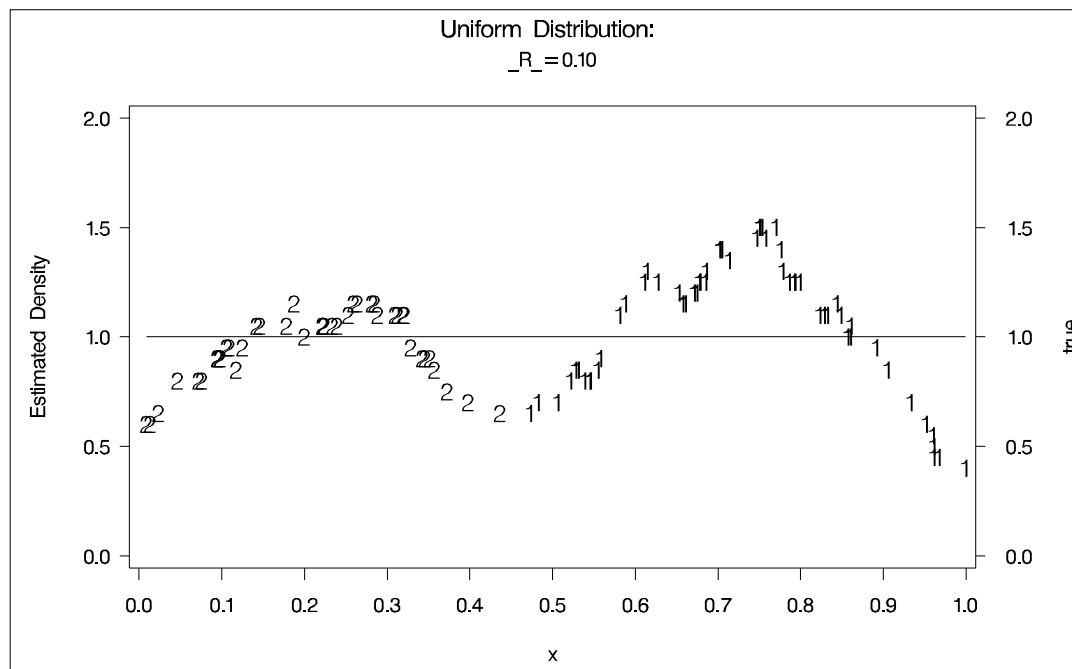
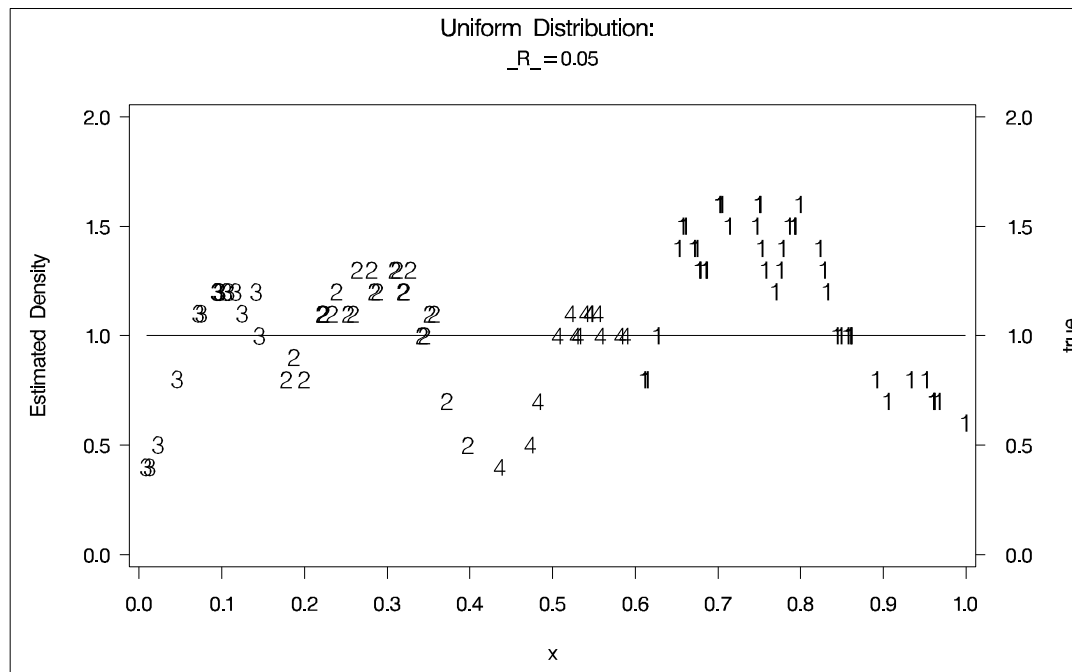


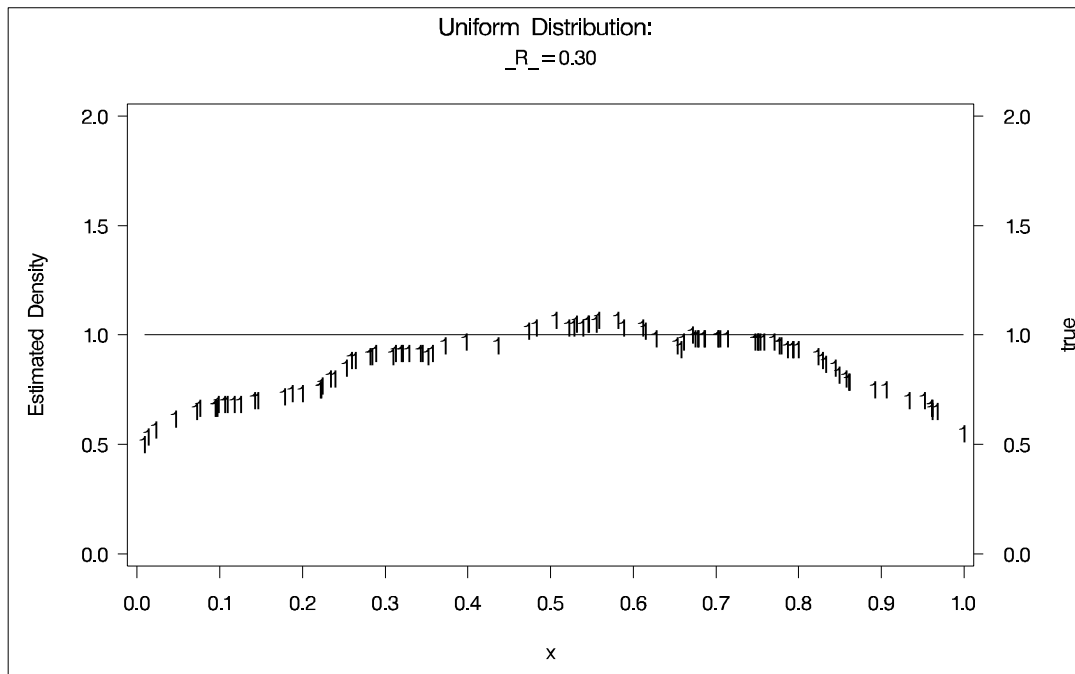
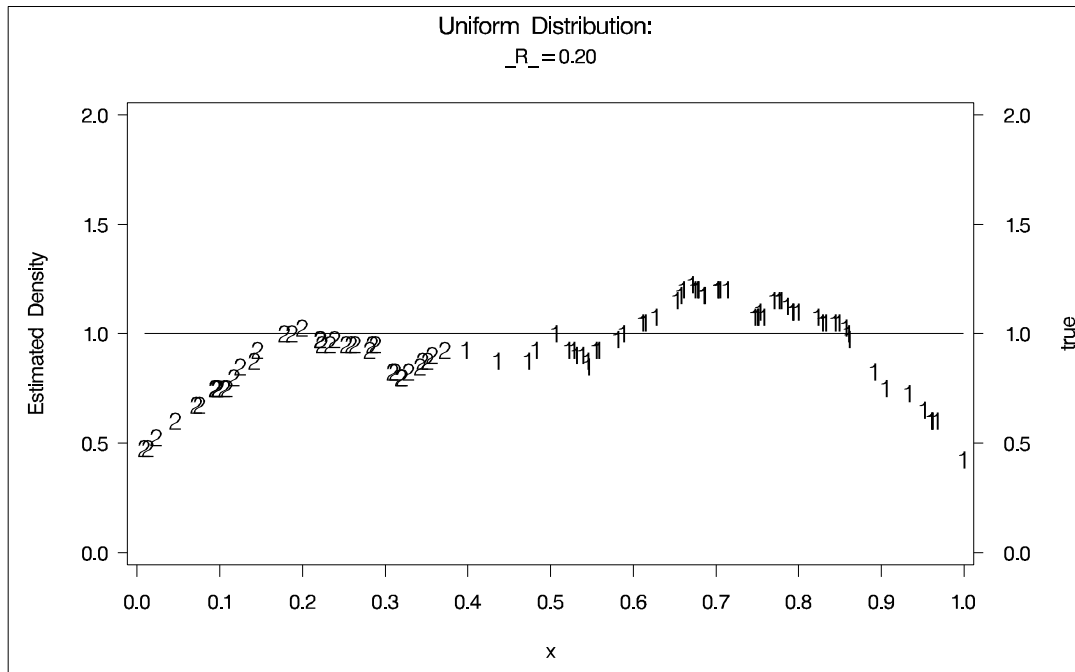


Modeclus Example with Univariate Distributions  
Uniform Distribution

The MODECLUS Procedure

	Cluster Summary	
R	Number of Clusters	Frequency of Unclassified Objects
0.05	4	0
0.1	2	0
0.2	2	0
0.3	1	0





The following statements produce Output 42.1.2:

```

title2 'Exponential Distribution';
data expon;
  drop n;
  do n=1 to 100;
    x=ranexp(123);
    true=exp(-x);
    output;
  end;

axis1 label=(angle=90 rotate=0) minor=none
      order=(0 to 2 by 0.5);
axis2 minor=none;

proc modeclus data=expon m=1 k=10 20 40 out=out short;
  var x;

proc gplot;
  plot density*x=cluster /frame cframe=ligr
                        vzero nolegend
                        vaxis=axis1 haxis=axis2;
  plot2 true*x=9/vaxis=axis1 haxis=axis2;
  by _K_;
run;
/*****/

proc modeclus data=expon m=1 r=.20 .40 .80 out=out short;
  var x;

axis1 label=(angle=90 rotate=0)
      minor=none order=(0 to 1 by 0.5);
proc gplot;
  plot density*x=cluster /frame cframe=ligr
                        vzero nolegend
                        vaxis=axis1 haxis=axis2;
  plot2 true*x=9/vaxis=axis1 haxis=axis2;
  by _R_;
run;
/*****/

title3 'Different Density-Estimation and Clustering Windows';
proc modeclus data=expon m=1 r=.20 ck=10 20 40
      out=out short;
  var x;

proc gplot;
  plot density*x=cluster /frame cframe=ligr
                        vzero nolegend
                        vaxis=axis1 haxis=axis2;
  plot2 true*x=9/vaxis=axis1 haxis=axis2;
  by _CK_;
run;
/*****/

```

```

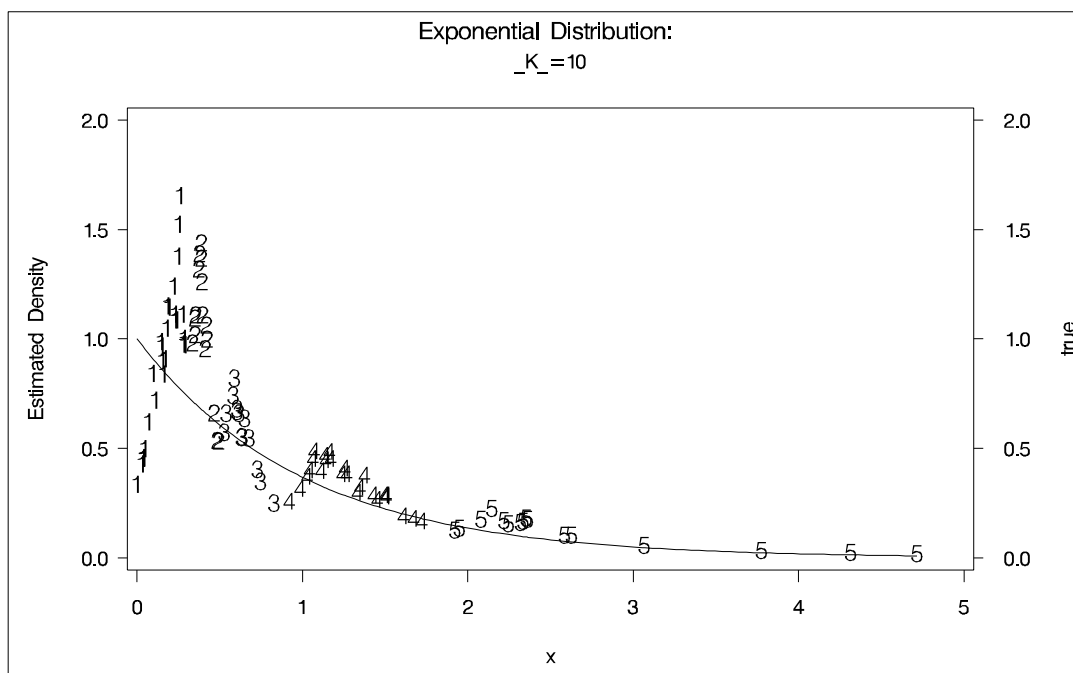
title3 'Cascaded Density Estimates Using Arithmetic Means';
proc modeclus data=expon m=1 r=.20 cascade=1 2 4 am out=out short;
  var x;

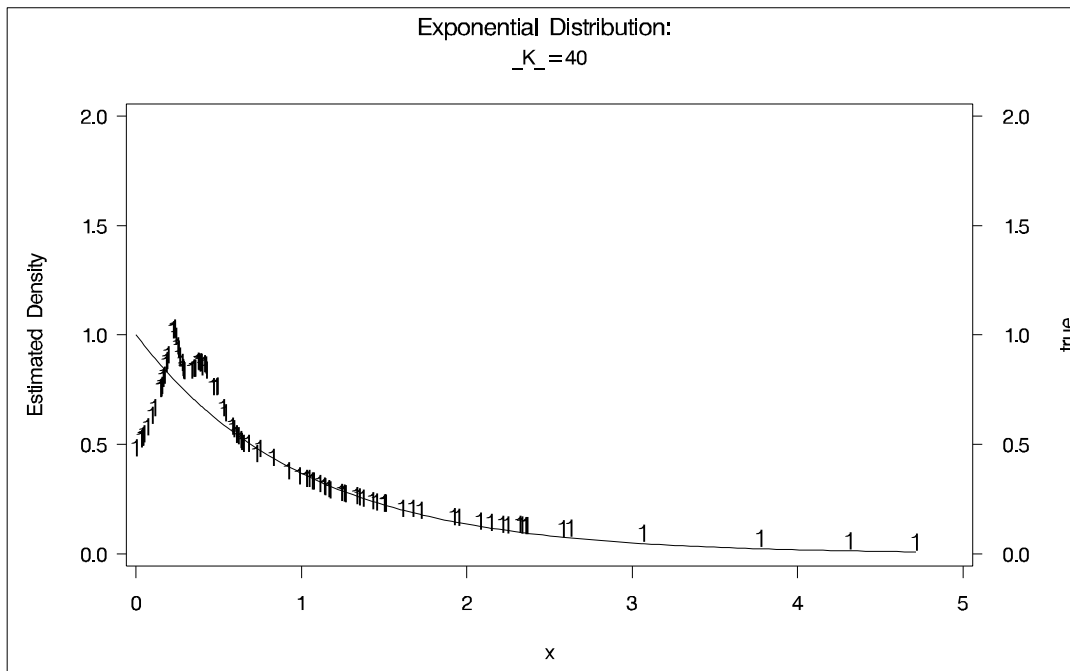
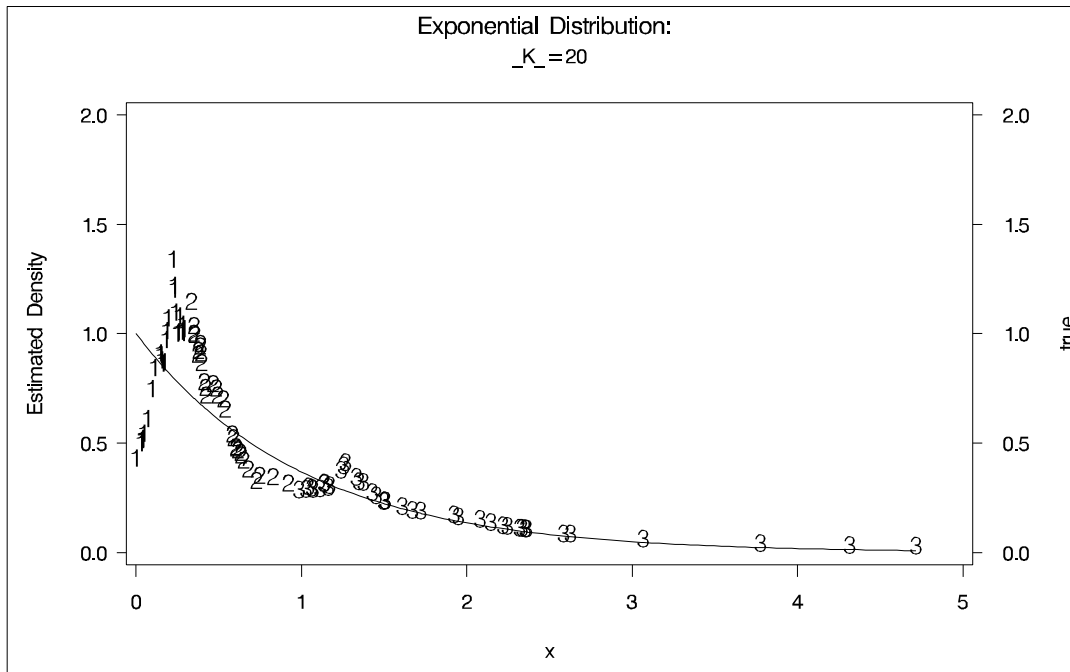
proc gplot;
  plot density*x=cluster /frame cframe=ligr
                                vzero nolegend
                                vaxis=axis1 haxis=axis2;
  plot2 true*x=9/vaxis=axis1 haxis=axis2;
  by _R_ _CASCAD_;
run;

```

**Output 42.1.2.** Cluster Analysis of Sample from an Exponential Distribution

Modeclus Example with Univariate Distributions Exponential Distribution		
The MODECLUS Procedure		
Cluster Summary		
K	Number of Clusters	Frequency of Unclassified Objects
10	5	0
20	3	0
40	1	0





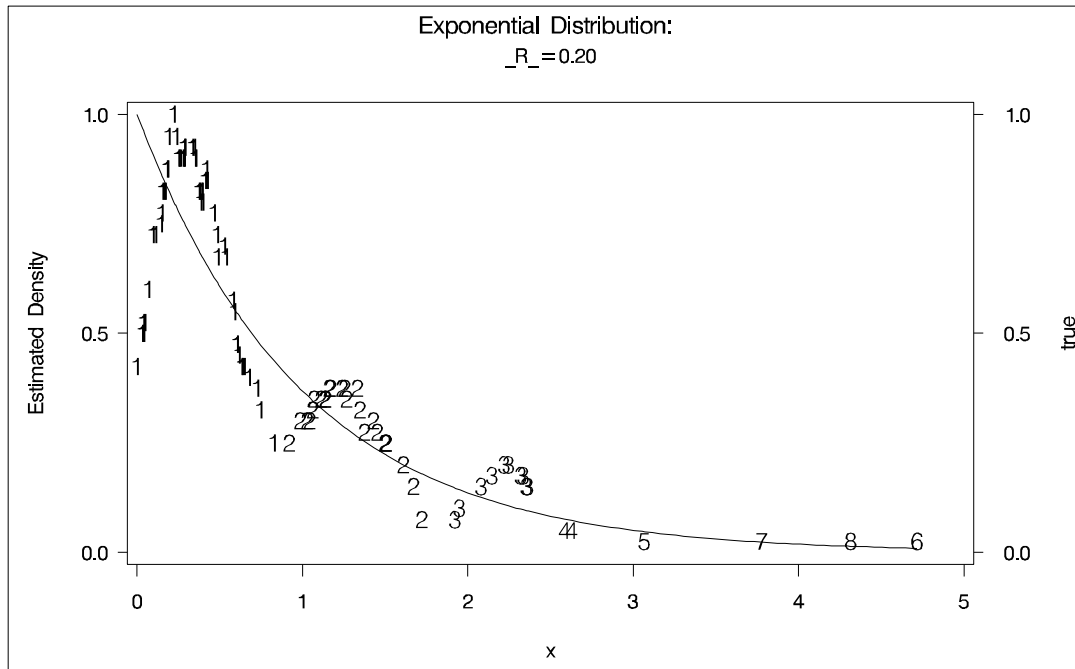
## Modeclus Example with Univariate Distributions

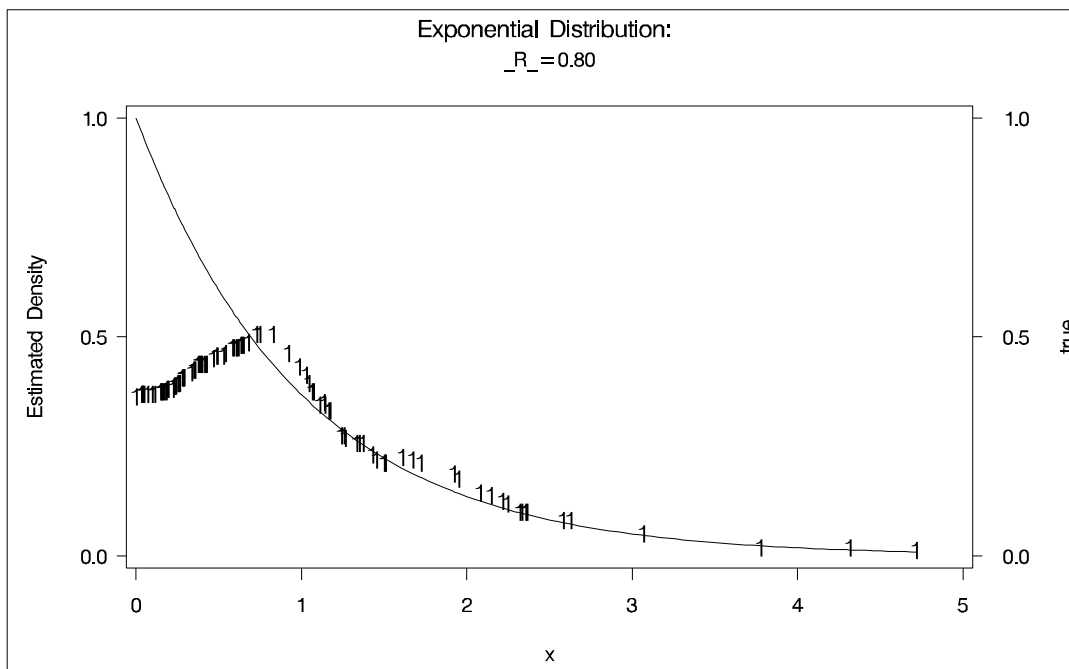
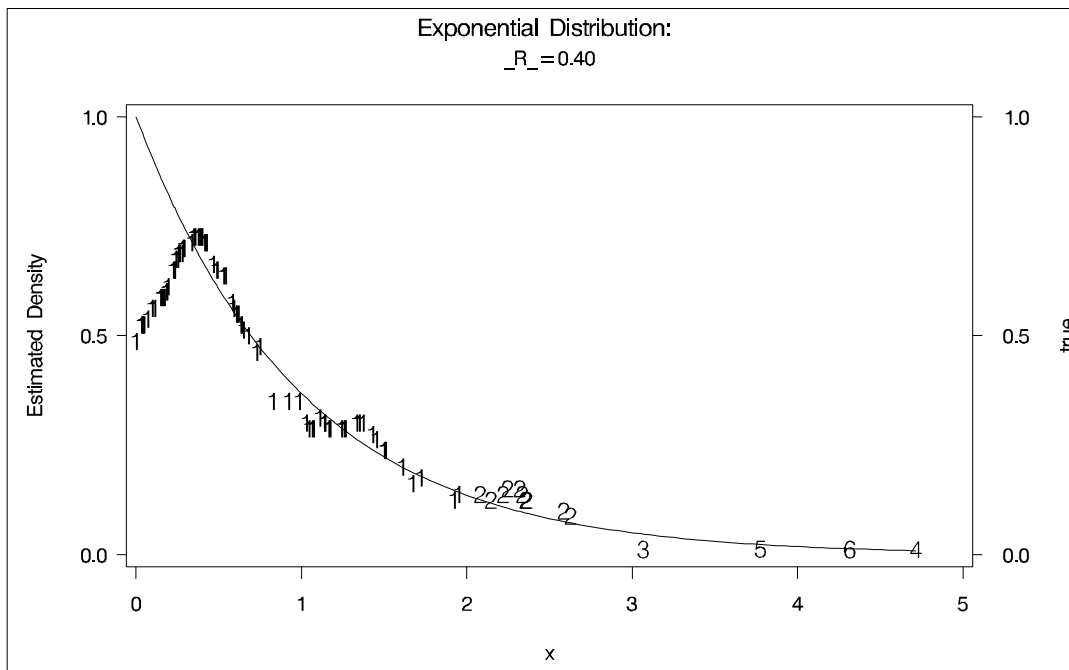
### Exponential Distribution

### The MODECLUS Procedure

### Cluster Summary

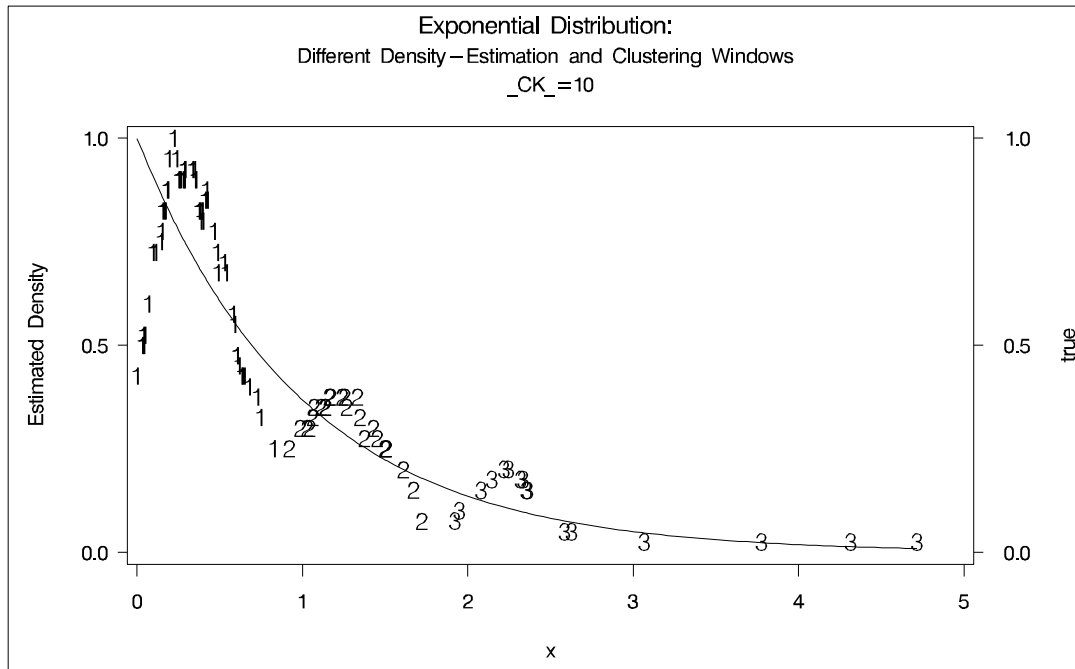
R	Number of Clusters	Frequency of Unclassified Objects
0.2	8	0
0.4	6	0
0.8	1	0

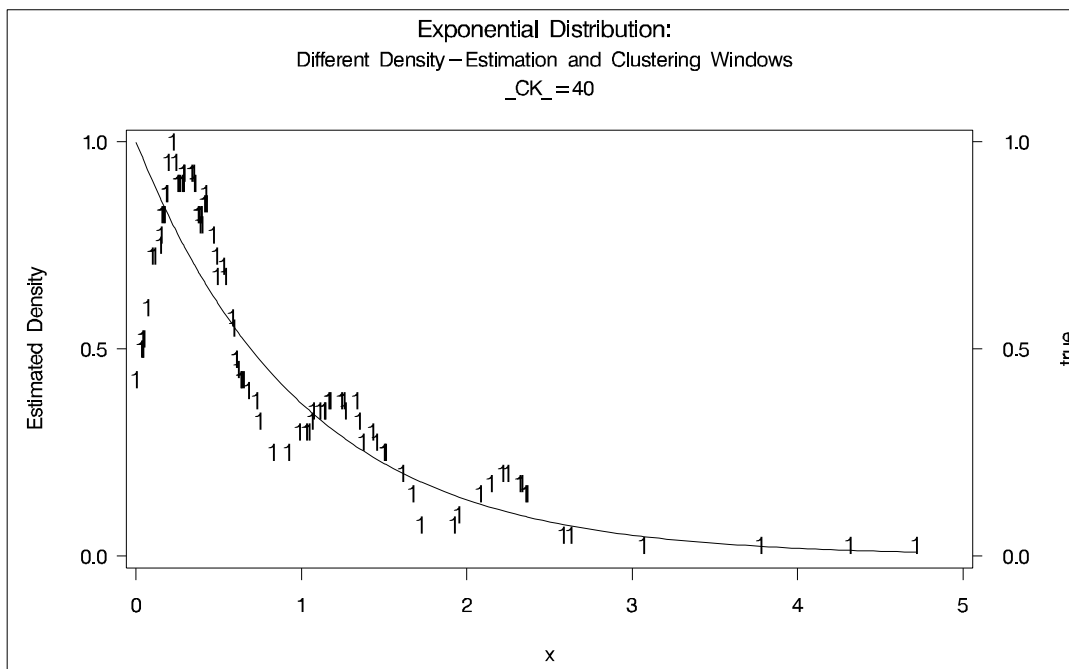
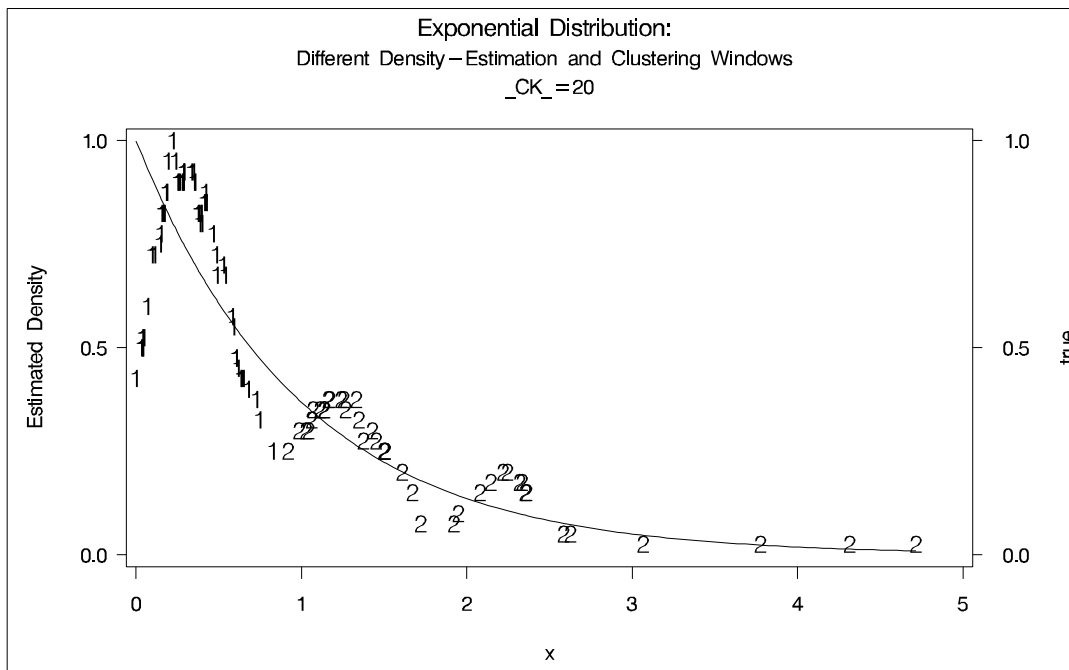




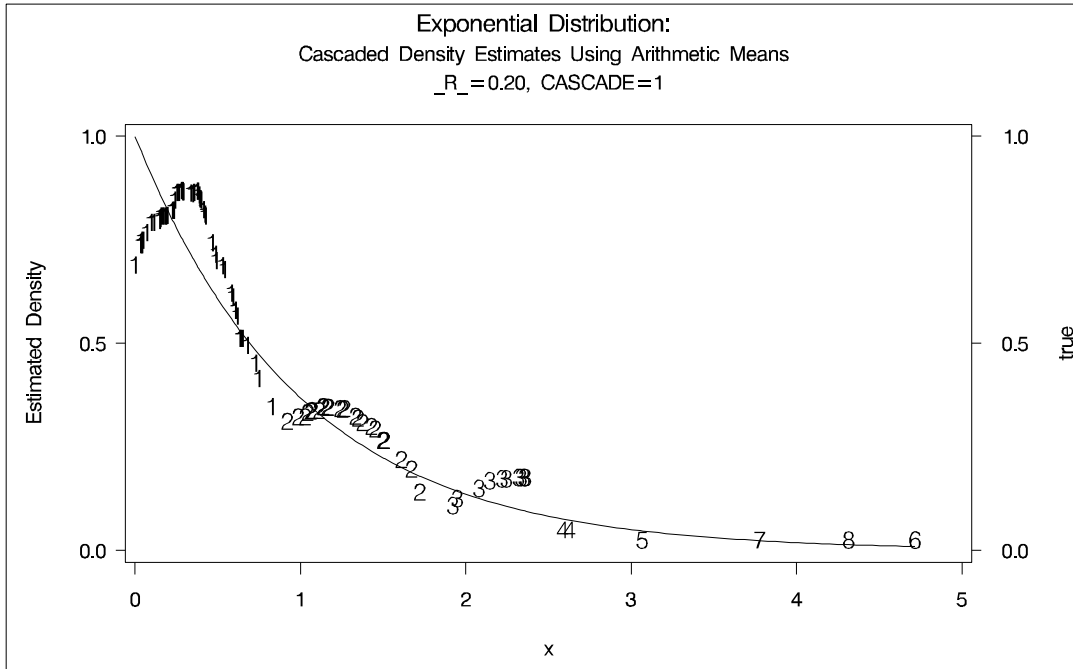


Modeclus Example with Different Density-Estimation and Clustering Windows			
The MODECLUS Procedure			
Cluster Summary			
R	CK	Number of Clusters	Frequency of Unclassified Objects
0.2	10	3	0
0.2	20	2	0
0.2	40	1	0





Modeclus Example with Cascaded Density Estimates Using Arithmetic Means			
The MODECLUS Procedure			
Cluster Summary			
	R	Cascade	Frequency of Unclassified Objects
-----			
	0.2	1	8
	0.2	2	8
	0.2	4	7
			0





The following statements produce Output 42.1.3:

```

title2 'Normal Mixture Distribution';
data normix;
  drop n sigma;
  sigma=.125;
  do n=1 to 100;
    x=rannor(456)*sigma+mod(n,2)/2;
    true=exp(-.5*(x/sigma)**2)+exp(-.5*((x-.5)/sigma)**2);
    true=.5*true/(sigma*sqrt(2*3.1415926536));
    output;
  end;

axis1 label=(angle=90 rotate=0) minor=none order=(0 to 3 by 0.5);
axis2 minor=none;

proc modeclus data=normix m=1 k=10 20 40 60 out=out short;
  var x;

proc gplot;
  plot density*x=cluster /frame cframe=ligr
                        vzero nolegend
                        vaxis=axis1 haxis=axis2;
  plot2 true*x=9/vaxis=axis1 haxis=axis2;
  by _K_;
run;
/*****/

proc modeclus data=normix m=1 r=.05 .10 .20 .30 out=out short;
  var x;

proc gplot;
  plot density*x=cluster /frame cframe=ligr
                        vzero nolegend
                        vaxis=axis1 haxis=axis2 ;
  plot2 true*x=9/vaxis=axis1 haxis=axis2 ;
  by _R_;
run;
/*****/

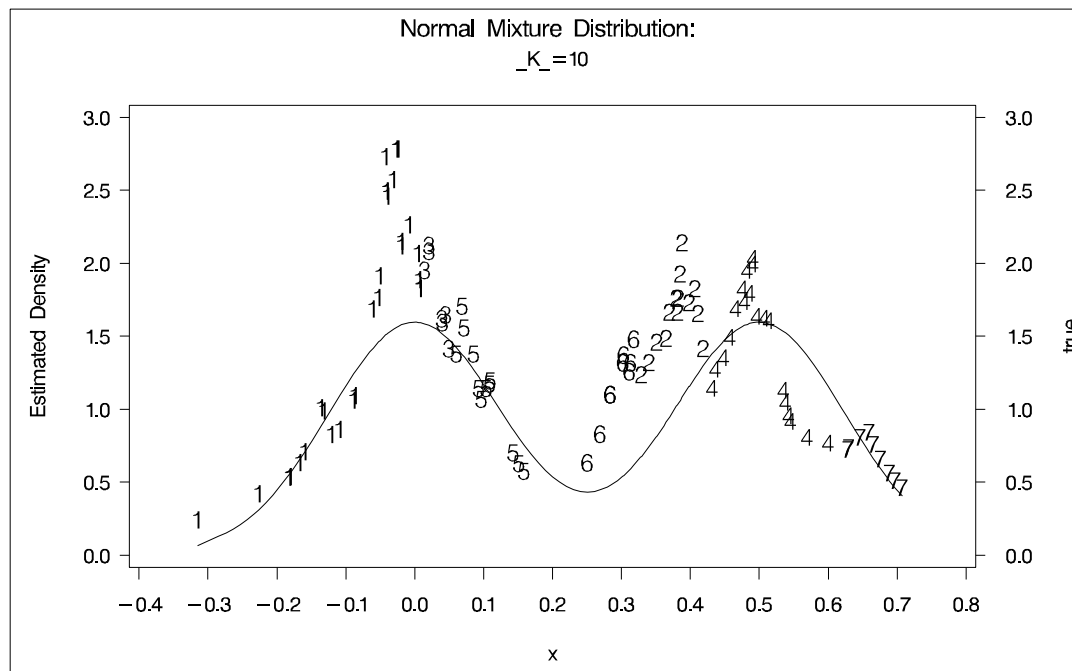
title3 'Cascaded Density Estimates Using Arithmetic Means';
proc modeclus data=normix m=1 r=.05 cascade=1 2 4 am out=out short;
  var x;

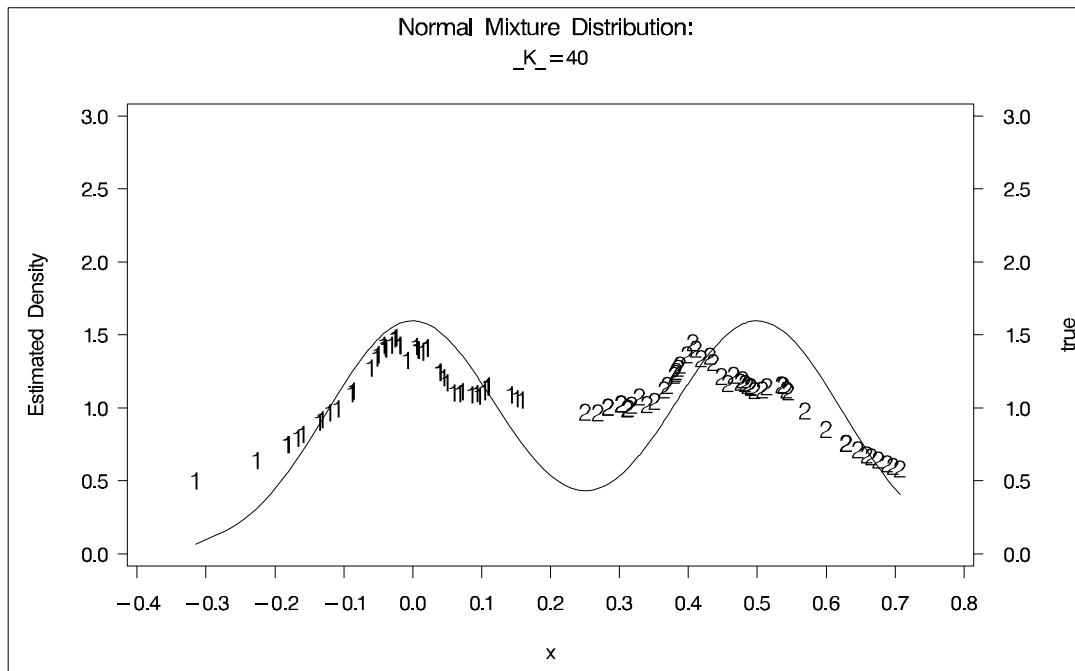
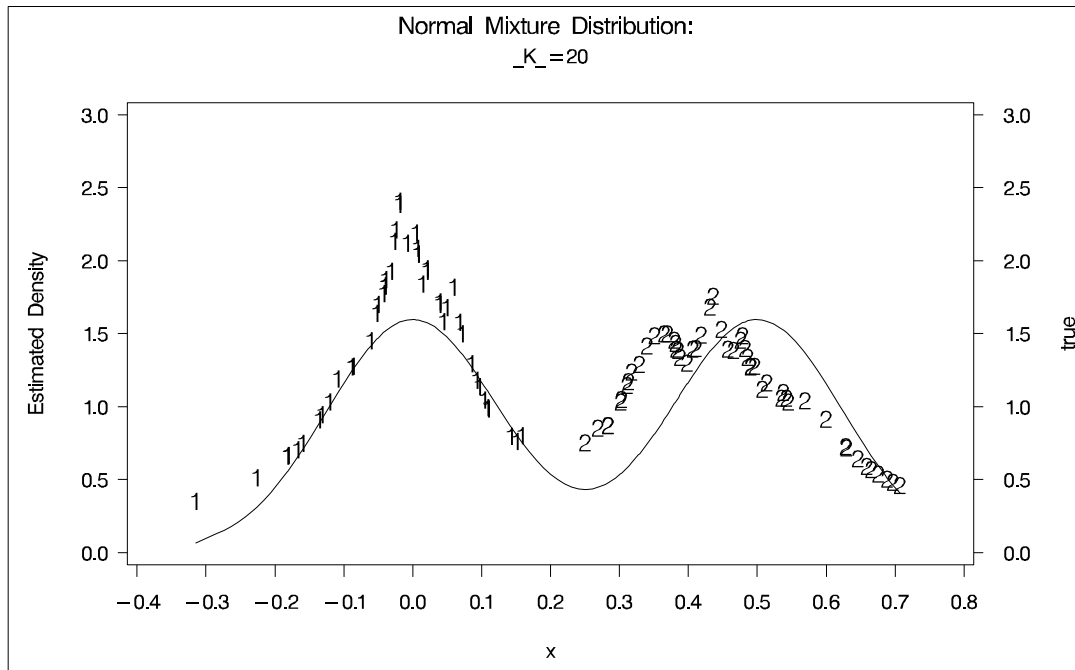
axis1 label=(angle=90 rotate=0)
      minor=none order=(0 to 2 by 0.5);
proc gplot;
  plot density*x=cluster /frame cframe=ligr
                        vzero nolegend
                        vaxis=axis1 haxis=axis2 ;
  plot2 true*x=9/vaxis=axis1 haxis=axis2 ;
  by _R_ _CASCAD_;
run;

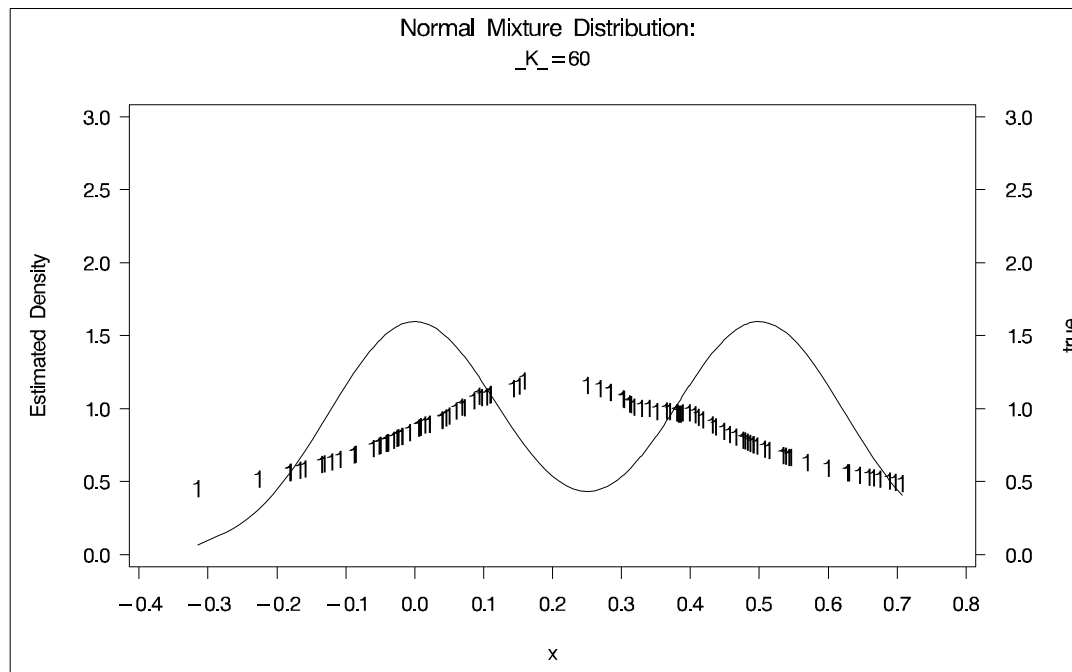
```

**Output 42.1.3.** Cluster Analysis of Sample from a Bimodal Mixture of Two Normal Distributions

Modeclus Example with Normal Mixture Distribution		
The MODECLUS Procedure		
Cluster Summary		
K	Number of Clusters	Frequency of Unclassified Objects
10	7	0
20	2	0
40	2	0
60	1	0







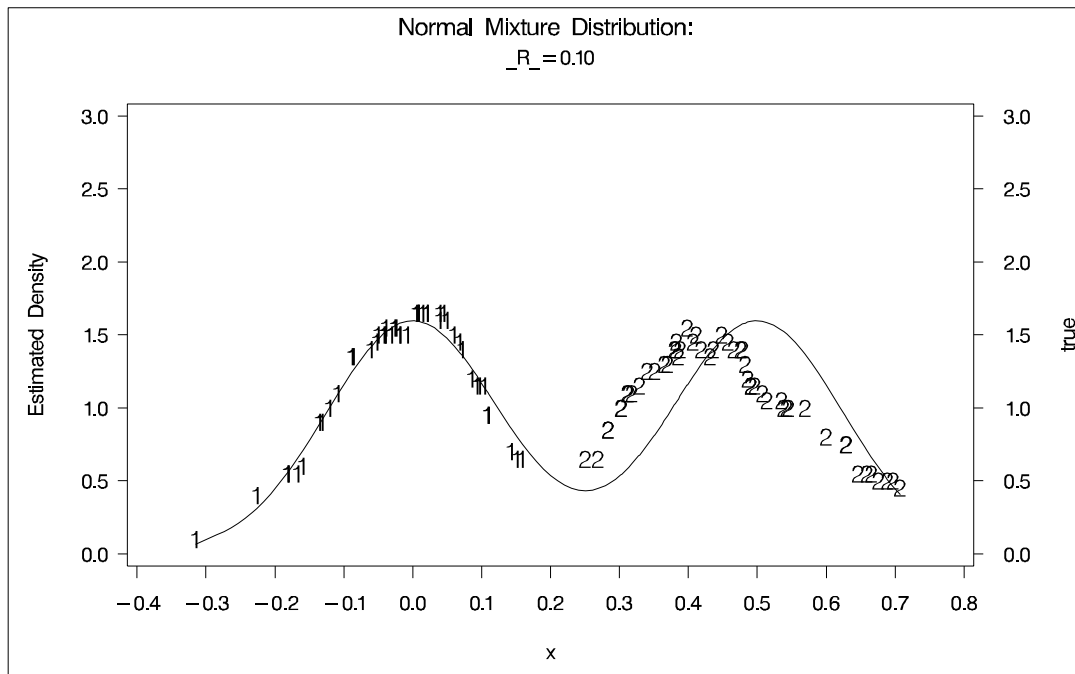
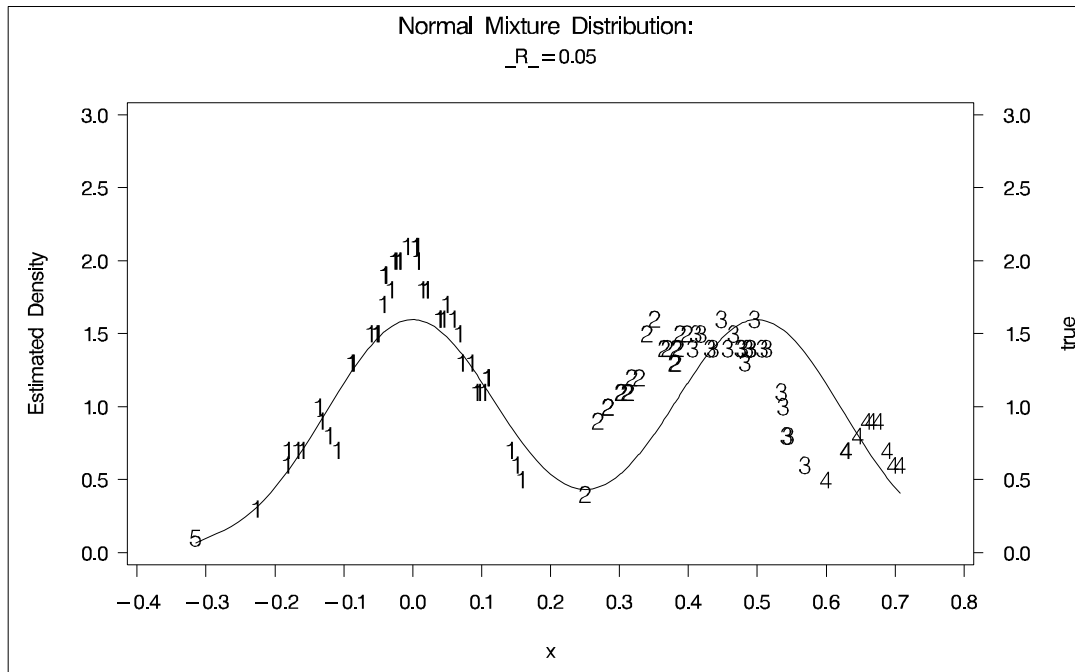
Modeclus Example with  
Normal Mixture Distribution

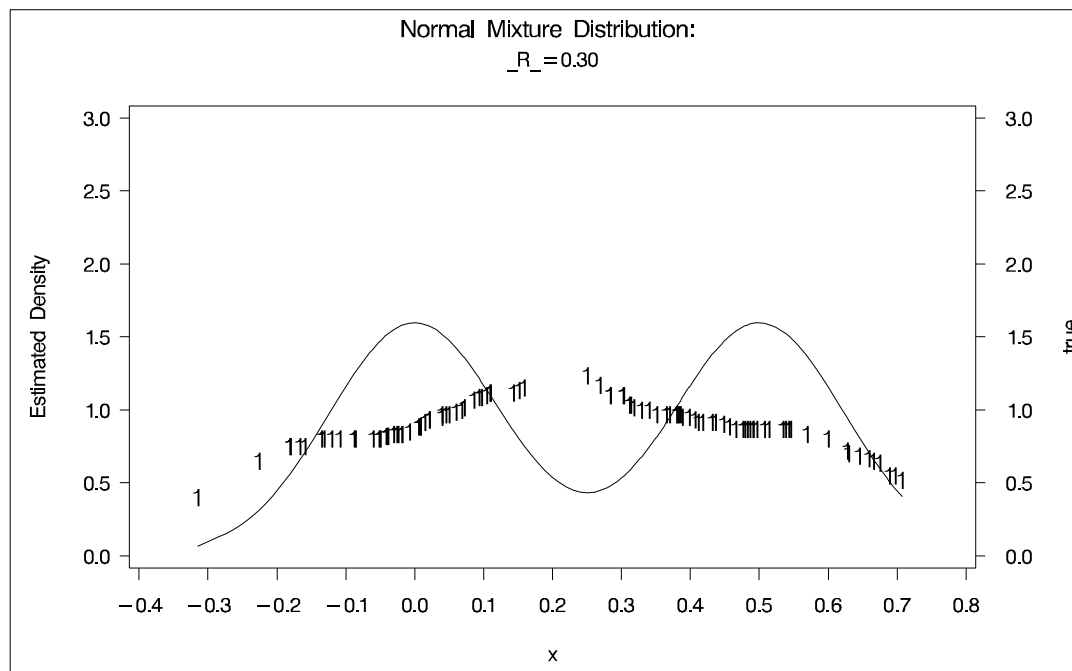
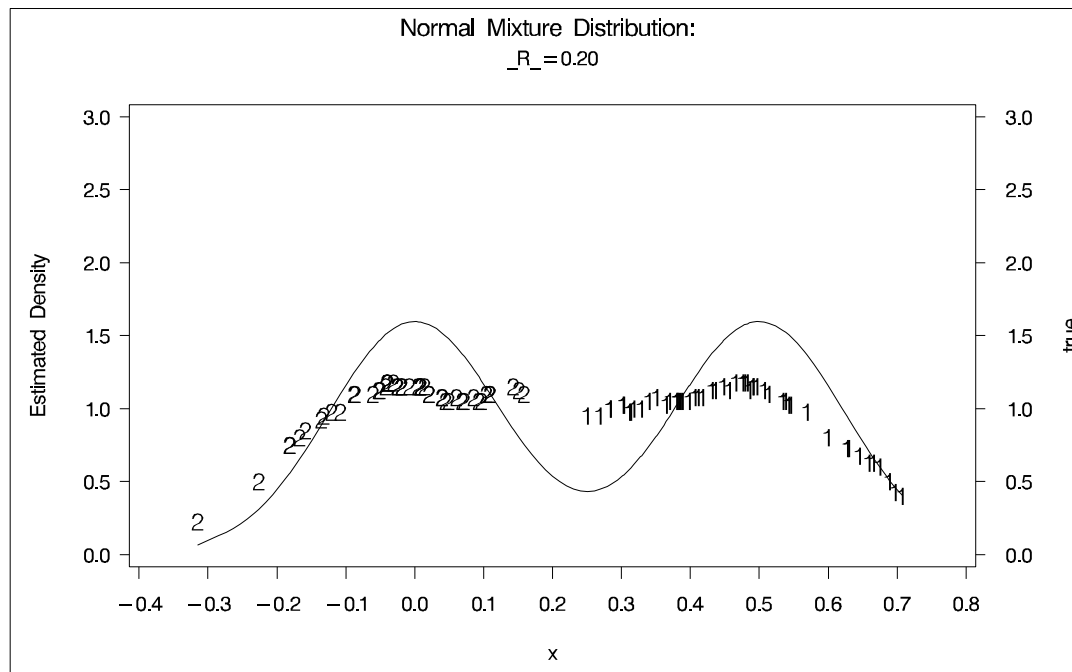
The MODECLUS Procedure

Cluster Summary

R	Number of Clusters	Frequency of Unclassified Objects
0.05	5	0
0.1	2	0
0.2	2	0
0.3	1	0





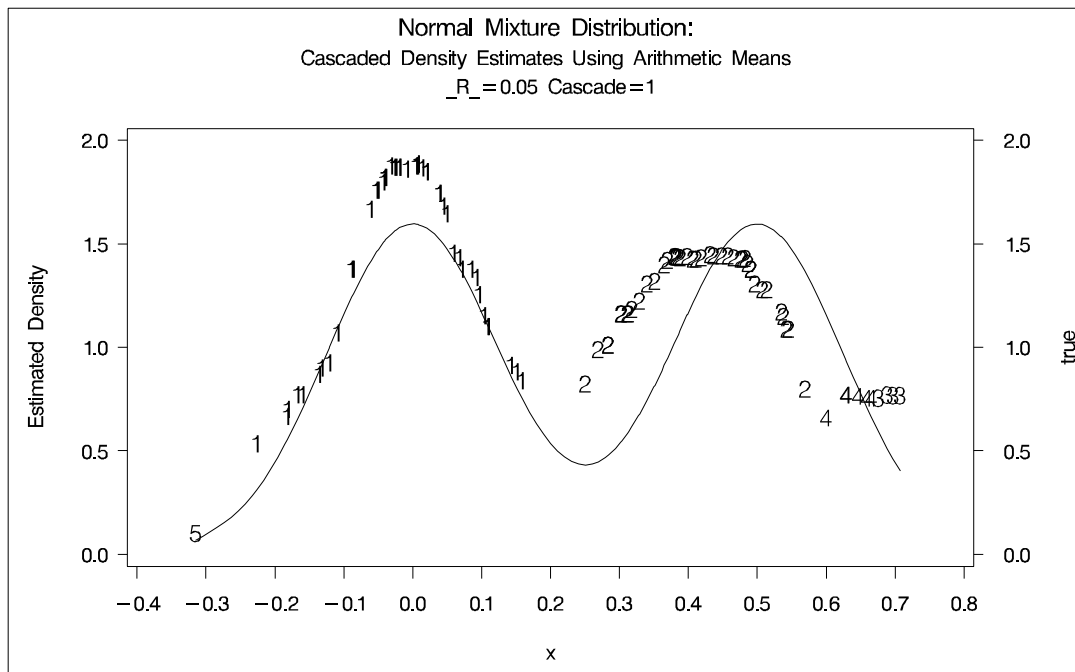


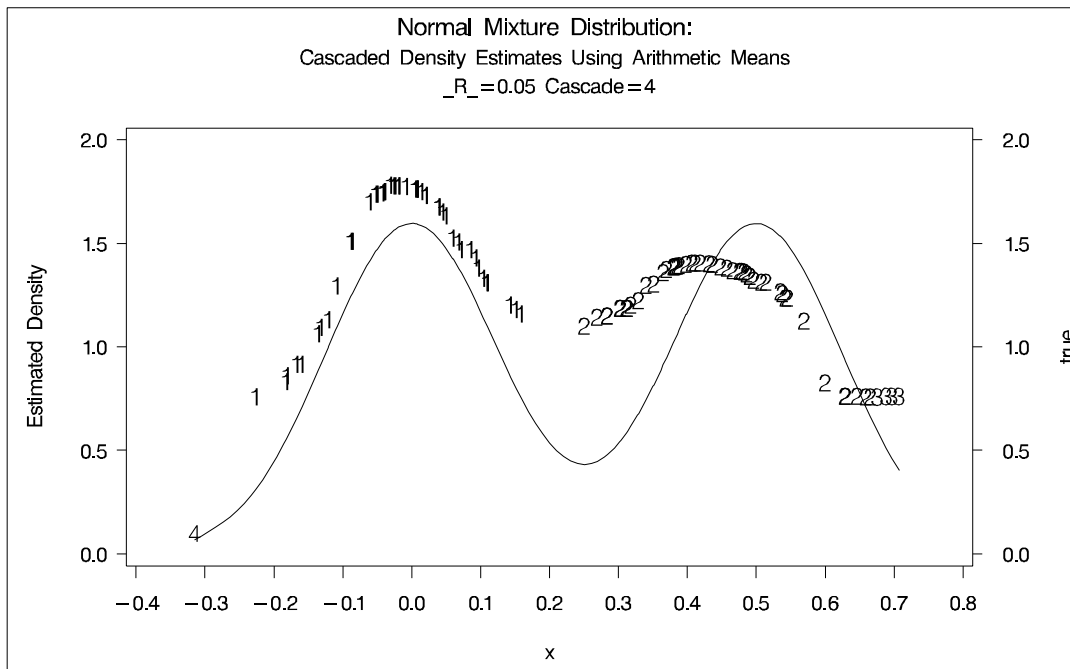
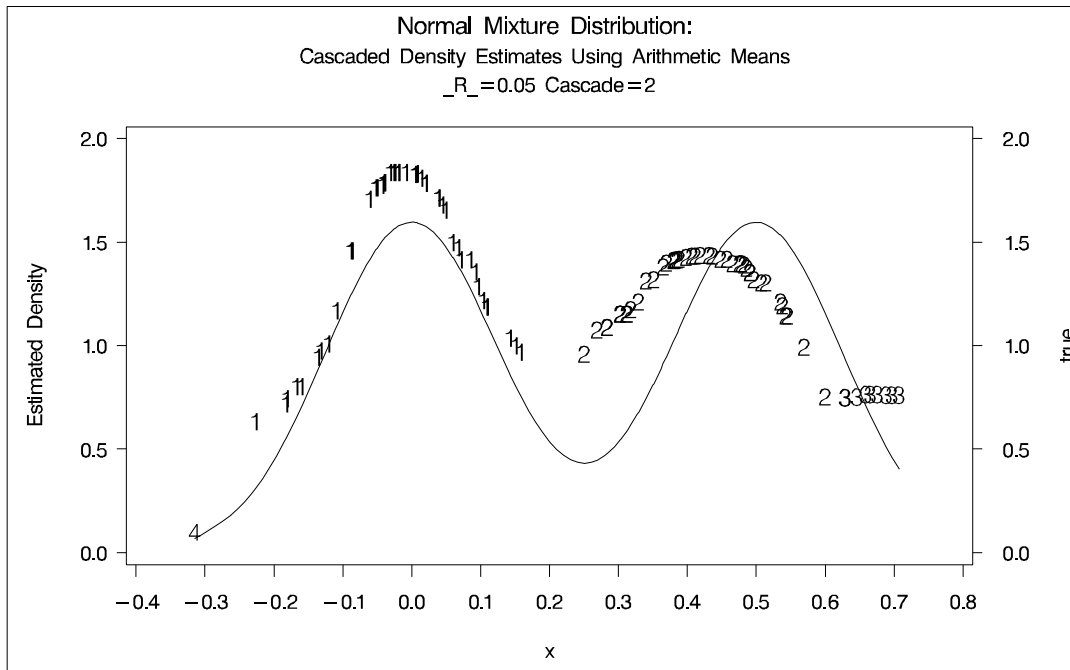
Modeclus Example with  
Normal Mixture Distribution  
Cascaded Density Estimates Using Arithmetic Means

The MODECLUS Procedure

Cluster Summary

R	Cascade	Number of Clusters	Frequency of Unclassified Objects
0.05	1	5	0
0.05	2	4	0
0.05	4	4	0





## Example 42.2. Cluster Analysis of Flying Mileages between Ten American Cities

This example uses distance data and illustrates the use of the TRANSPOSE procedure and the DATA step to fill in the upper triangle of the distance matrix. The results are displayed in Output 42.2.1 through Output 42.2.2.

The following statements produce Output 42.2.1:

```

title 'Modeclus Analysis of 10 American Cities';
title2 'Based on Flying Mileages';
options ls=90;

data mileages(type=distance);
  input (ATLANTA CHICAGO DENVER HOUSTON LOSANGELES
        MIAMI NEWYORK SANFRAN SEATTLE WASHDC) (5.)
        @53 CITY $15.;
  datalines;
    0
    587    0
    1212  920    0
    701  940  879    0
    1936 1745  831 1374    0
    604 1188 1726  968 2339    0
    748  713 1631 1420 2451 1092    0
    2139 1858  949 1645  347 2594 2571    0
    2182 1737 1021 1891  959 2734 2408  678    0
    543  597 1494 1220 2300  923  205 2442 2329    0
    ATLANTA
    CHICAGO
    DENVER
    HOUSTON
    LOS ANGELES
    MIAMI
    NEW YORK
    SAN FRANCISCO
    SEATTLE
    WASHINGTON D.C.
  ;

*-----Fill in Upper Triangle of Distance Matrix-----;
proc transpose out=tran;
  copy CITY;
data mileages(type=distance);
  merge mileages tran;
  array var ATLANTA--WASHDC;
  array col coll-coll10;
  drop coll-coll10 _name_;
  do over var;
    var=sum(var,col);
  end;

*-----Clustering with K-Nearest-Neighbor Density Estimates-----;
proc modeclus data=mileages all m=1 k=3;
  id CITY;
run;

```

**Output 42.2.1.** Clustering with K-Nearest-Neighbor Density Estimates

Modeclus Analysis of 10 American Cities Based on Flying Mileages		
The MODECLUS Procedure		
Nearest Neighbor List		
CITY	Neighbor	Distance
ATLANTA	WASHINGTON D.C.	543.0000000
	CHICAGO	587.0000000
CHICAGO	ATLANTA	587.0000000
	WASHINGTON D.C.	597.0000000
DENVER	LOS ANGELES	831.0000000
	HOUSTON	879.0000000
HOUSTON	ATLANTA	701.0000000
	DENVER	879.0000000
LOS ANGELES	SAN FRANCISCO	347.0000000
	DENVER	831.0000000
MIAMI	ATLANTA	604.0000000
	WASHINGTON D.C.	923.0000000
NEW YORK	WASHINGTON D.C.	205.0000000
	CHICAGO	713.0000000
SAN FRANCISCO	LOS ANGELES	347.0000000
	SEATTLE	678.0000000
SEATTLE	SAN FRANCISCO	678.0000000
	LOS ANGELES	959.0000000
WASHINGTON D.C.	NEW YORK	205.0000000
	ATLANTA	543.0000000

Modeclus Analysis of 10 American Cities Based on Flying Mileages						
The MODECLUS Procedure K=3 METHOD=1						
Sums of Density Estimates Within Neighborhood						
Cluster	CITY	Estimated Density	Same Cluster	Other Clusters	Total	Cluster Proportion Same/Total
1	ATLANTA	0.00025554	0.0005275	0	0.0005275	1.000
	CHICAGO	0.00025126	0.00053178	0	0.00053178	1.000
	HOUSTON	0.00017065	0.00025554	0.00017065	0.00042619	0.600
	MIAMI	0.00016251	0.00053178	0	0.00053178	1.000
	NEW YORK	0.00021038	0.0005275	0	0.0005275	1.000
	WASHINGTON D.C.	0.00027624	0.00046592	0	0.00046592	1.000
2	DENVER	0.00017065	0.00018051	0.00017065	0.00035115	0.514
	LOS ANGELES	0.00018051	0.00039189	0	0.00039189	1.000
	SAN FRANCISCO	0.00022124	0.00033692	0	0.00033692	1.000
	SEATTLE	0.00015641	0.00040174	0	0.00040174	1.000
Boundary Objects						
CITY		Density	-Cluster Proportions- Cluster	1	2	
DENVER		0.0001706485	2	0.486	0.514	
HOUSTON		0.0001706485	1	0.600	0.400	
Cluster Statistics						
Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density		
1	6	0.00027624	1	0.00017065		
2	4	0.00022124	1	0.00017065		

Modeclus Analysis of 10 American Cities Based on Flying Mileages					
The MODECLUS Procedure					
Cluster Summary					
K	Number of Clusters	Frequency of Unclassified Objects			
3	2	0			

The following statements produce Output 42.2.2:

```
*-----Clustering with Uniform Kernel Density Estimates-----;
proc modeclus data=mileages all m=1 r=600 800;
  id CITY;
run;
```

**Output 42.2.2.** Clustering with Uniform Kernel Density Estimates

Modeclus Analysis of 10 American Cities Based on Flying Mileages		
The MODECLUS Procedure		
CITY	Nearest Neighbor List Neighbor	Distance
ATLANTA	WASHINGTON D.C.	543.0000000
	CHICAGO	587.0000000
	MIAMI	604.0000000
	HOUSTON	701.0000000
	NEW YORK	748.0000000
CHICAGO	ATLANTA	587.0000000
	WASHINGTON D.C.	597.0000000
	NEW YORK	713.0000000
HOUSTON	ATLANTA	701.0000000
LOS ANGELES	SAN FRANCISCO	347.0000000
MIAMI	ATLANTA	604.0000000
NEW YORK	WASHINGTON D.C.	205.0000000
	CHICAGO	713.0000000
	ATLANTA	748.0000000
SAN FRANCISCO	LOS ANGELES	347.0000000
	SEATTLE	678.0000000
SEATTLE	SAN FRANCISCO	678.0000000
WASHINGTON D.C.	NEW YORK	205.0000000
	ATLANTA	543.0000000
	CHICAGO	597.0000000



Modeclus Analysis of 10 American Cities  
Based on Flying Mileages

The MODECLUS Procedure  
R=600 METHOD=1

Sums of Density Estimates Within Neighborhood

Cluster	CITY	Estimated Density	Same Cluster	Other Clusters	Total	Cluster Proportion Same/Total
1	ATLANTA	0.00025	0.00058333	0	0.00058333	1.000
	CHICAGO	0.00025	0.00058333	0	0.00058333	1.000
	NEW YORK	0.00016667	0.00033333	0	0.00033333	1.000
	WASHINGTON D.C.	0.00033333	0.00066667	0	0.00066667	1.000
2	LOS ANGELES	0.00016667	0.00016667	0	0.00016667	1.000
	SAN FRANCISCO	0.00016667	0.00016667	0	0.00016667	1.000
3	DENVER	0.00008333	0	0	0	.
4	HOUSTON	0.00008333	0	0	0	.
5	MIAMI	0.00008333	0	0	0	.
6	SEATTLE	0.00008333	0	0	0	.

No Boundary Objects

Cluster Statistics

Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density
1	4	0.00033333	0	.
2	2	0.00016667	0	.
3	1	0.00008333	0	.
4	1	0.00008333	0	.
5	1	0.00008333	0	.
6	1	0.00008333	0	.

Modeclus Analysis of 10 American Cities  
Based on Flying Mileages

The MODECLUS Procedure  
R=800 METHOD=1

Sums of Density Estimates Within Neighborhood

Cluster	CITY	Estimated Density	Same Cluster	Other Clusters	Total	Cluster Proportion Same/Total
1	ATLANTA	0.000375	0.001	0	0.001	1.000
	CHICAGO	0.00025	0.000875	0	0.000875	1.000
	HOUSTON	0.000125	0.000375	0	0.000375	1.000
	MIAMI	0.000125	0.000375	0	0.000375	1.000
	NEW YORK	0.00025	0.000875	0	0.000875	1.000
	WASHINGTON D.C.	0.00025	0.000875	0	0.000875	1.000
2	LOS ANGELES	0.000125	0.0001875	0	0.0001875	1.000
	SAN FRANCISCO	0.0001875	0.00025	0	0.00025	1.000
	SEATTLE	0.000125	0.0001875	0	0.0001875	1.000
3	DENVER	0.0000625	0	0	0	.

No Boundary Objects

Cluster Statistics

Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density
1	6	0.000375	0	.
2	3	0.0001875	0	.
3	1	0.0000625	0	.

Modeclus Analysis of 10 American Cities Based on Flying Mileages		
The MODECLUS Procedure		
Cluster Summary		
R	Number of Clusters	Frequency of Unclassified Objects
600	6	0
800	3	0

The following statements produce Output 42.2.3:

```
*-----Uniform Kernel Density Estimates, Clustering
      Neighborhoods extended to nearest neighbor-----;
proc modeclus data=mileages list m=1 ck=2 r=600 800;
    id CITY;
run;
```

**Output 42.2.3.** Uniform Kernel Density Estimates, Clustering Neighborhoods Extended to Nearest Neighbor

Modeclus Analysis of 10 American Cities Based on Flying Mileages						
The MODECLUS Procedure						
CK=2 R=600 METHOD=1						
Sums of Density Estimates Within Neighborhood						
Cluster	CITY	Estimated Density	Same Cluster	Other Clusters	Total	Cluster Proportion Same/Total
1	ATLANTA	0.00025	0.00058333	0	0.00058333	1.000
	CHICAGO	0.00025	0.00058333	0	0.00058333	1.000
	HOUSTON	0.00008333	0.00025	0	0.00025	1.000
	MIAMI	0.00008333	0.00025	0	0.00025	1.000
	NEW YORK	0.00016667	0.00033333	0	0.00033333	1.000
	WASHINGTON D.C.	0.00033333	0.00066667	0	0.00066667	1.000
2	DENVER	0.00008333	0.00016667	0	0.00016667	1.000
	LOS ANGELES	0.00016667	0.00016667	0	0.00016667	1.000
	SAN FRANCISCO	0.00016667	0.00016667	0	0.00016667	1.000
	SEATTLE	0.00008333	0.00016667	0	0.00016667	1.000
Cluster Statistics						
Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density		
1	6	0.00033333	0	.		
2	4	0.00016667	0	.		

Modeclus Analysis of 10 American Cities Based on Flying Mileages						
The MODECLUS Procedure						
CK=2 R=800 METHOD=1						
Sums of Density Estimates Within Neighborhood						
Cluster	CITY	Estimated Density	Same Cluster	Other Clusters	Total	Cluster Proportion Same/Total
1	ATLANTA	0.000375	0.001	0	0.001	1.000
	CHICAGO	0.00025	0.000875	0	0.000875	1.000
	HOUSTON	0.000125	0.000375	0	0.000375	1.000
	MIAMI	0.000125	0.000375	0	0.000375	1.000
	NEW YORK	0.00025	0.000875	0	0.000875	1.000
	WASHINGTON D.C.	0.00025	0.000875	0	0.000875	1.000
2	DENVER	0.0000625	0.000125	0	0.000125	1.000
	LOS ANGELES	0.000125	0.0001875	0	0.0001875	1.000
	SAN FRANCISCO	0.0001875	0.00025	0	0.00025	1.000
	SEATTLE	0.000125	0.0001875	0	0.0001875	1.000
Cluster Statistics						
Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density		
1	6	0.000375	0	.		
2	4	0.0001875	0	.		

Modeclus Analysis of 10 American Cities Based on Flying Mileages				
The MODECLUS Procedure				
Cluster Summary				
R	CK	Number of Clusters	Frequency of Unclassified Objects	
600	2	2	0	
800	2	2	0	

### Example 42.3. Cluster Analysis with Significance Tests

This example uses artificial data containing two clusters. One cluster is from a circular bivariate normal distribution. The other is a ring-shaped cluster that completely surrounds the first cluster. Without significance tests, the ring is divided into several sample clusters for any degree of smoothing that yields reasonable density estimates. The JOIN= option puts the ring back together. Output 42.3.1 displays a short summary generated from the first PROC MODECLUS statement. Output 42.3.2 contains a series of tables produced from the second PROC MODECLUS statement. The lack of *p*-value in the JOIN= option makes joining continue until only one cluster remains (see the description of the JOIN= option on page 2240). The cluster memberships are then plotted as displayed in Output 42.3.3.

```

title 'Modeclus Analysis with the JOIN= option';
title2 'A Normal Cluster Surrounded by a Ring Cluster';
options ls=120 ps=38;

data circle; keep x y;
  c=1;
  do n=1 to 30;
    x=rannor(5);
    y=rannor(5);
    output;
  end;

  c=2;
  do n=1 to 300;
    x=rannor(5);
    y=rannor(5);
    z=rannor(5)+8;
    l=z/sqrt(x**2+y**2);
    x=x*l;
    y=y*l;
    output;
  end;

axis1 label=(angle=90 rotate=0) minor=none
      order=(-10 to 10 by 5);
axis2 minor=none order=(-15 to 15 by 5);

proc modeclus data=circle m=1 r=1 to 3.5 by .25 join=20 short;
proc modeclus data=circle m=1 r=2.5 join out=out;

proc gplot data=out;
  plot y*x=cluster/frame cframe=ligr
      vzero nolegend
      vaxis=axis1 haxis=axis2 ;
  by _NJOIN_;
run;

```

**Output 42.3.1.** Significance Tests with the JOIN=20 and SHORT Options

Modeclus Analysis with the JOIN= option A Normal Cluster Surrounded by a Ring Cluster				
The MODECLUS Procedure				
Cluster Summary				
R	Number of Clusters Joined	Maximum P-value	Number of Clusters	Frequency of Unclassified Objects
1	36	0.9339	1	301
1.25	20	0.7131	1	301
1.5	10	0.3296	1	300
1.75	5	0.1990	2	0
2	5	0.0683	2	0
2.25	3	0.0504	2	0
2.5	4	0.0301	2	0
2.75	3	0.0585	2	0
3	5	0.0003	1	0
3.25	4	0.1923	2	0
3.5	4	0.0000	1	0

**Output 42.3.2.** Significance Tests with the JOIN Option

Modeclus Analysis with the JOIN= option A Normal Cluster Surrounded by a Ring Cluster									
The MODECLUS Procedure R=2.5 METHOD=1									
Cluster Statistics									
Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density	Mode Count	Saddle Count	Saddle Test: Version 92.7-- Overlap Count	Z	Approx P-value
1	103	0.00617328	22	0.00308664	39	19	0	2.495	0.5055
2	71	0.00571029	20	0.0043213	36	27	9	1.193	0.999
3	53	0.00509296	18	0.00401263	32	25	10	0.986	0.9999
4	45	0.00478429	19	0.00354964	30	22	14	1.429	0.9924
5	30	0.00462996	0	.	29	0	.	3.611	0.0301
6	28	0.00370397	17	0.00354964	23	22	9	0.000	1

Modeclus Analysis with the JOIN= option A Normal Cluster Surrounded by a Ring Cluster									
The MODECLUS Procedure R=2.5 METHOD=1									
Cluster Statistics									
Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density	Mode Count	Saddle Count	Saddle Test: Version 92.7-- Overlap Count	Z	Approx P-value
1	103	0.00617328	22	0.00308664	39	19	0	2.495	0.5055
2	71	0.00571029	20	0.0043213	36	27	9	1.193	0.999
3	53	0.00509296	18	0.00401263	32	25	10	0.986	0.9999
4	73	0.00478429	13	0.00293231	30	18	0	1.588	0.9778
5	30	0.00462996	0	.	29	0	.	3.611	0.0301

Modeclus Analysis with the JOIN= option  
A Normal Cluster Surrounded by a Ring Cluster

The MODECLUS Procedure  
R=2.5 METHOD=1

Cluster Statistics									
Cluster	Frequency	Maximum	Boundary	Estimated	-----Saddle Test: Version 92.7-----				
		Estimated Density		Saddle Density	Mode Count	Saddle Count	Overlap Count	Z	Approx P-value
1	156	0.00617328	17	0.00246931	39	15	0	3.130	0.1318
2	71	0.00571029	20	0.0043213	36	27	9	1.193	0.999
3	73	0.00478429	13	0.00293231	30	18	0	1.588	0.9778
4	30	0.00462996	0	.	29	0	.	3.611	0.0301

Modeclus Analysis with the JOIN= option  
A Normal Cluster Surrounded by a Ring Cluster

The MODECLUS Procedure  
R=2.5 METHOD=1

Cluster Statistics									
Cluster	Frequency	Maximum	Boundary	Estimated	-----Saddle Test: Version 92.7-----				
		Estimated Density		Saddle Density	Mode Count	Saddle Count	Overlap Count	Z	Approx P-value
1	156	0.00617328	17	0.00246931	39	15	0	3.130	0.1318
2	144	0.00571029	14	0.00293231	36	18	0	2.313	0.6447
3	30	0.00462996	0	.	29	0	.	3.611	0.0301

Modeclus Analysis with the JOIN= option  
A Normal Cluster Surrounded by a Ring Cluster

The MODECLUS Procedure  
R=2.5 METHOD=1

Cluster Statistics									
Cluster	Frequency	Maximum	Boundary	Estimated	-----Saddle Test: Version 92.7-----				
		Estimated Density		Saddle Density	Mode Count	Saddle Count	Overlap Count	Z	Approx P-value
1	300	0.00617328	0	.	39	0	.	4.246	0.0026
2	30	0.00462996	0	.	29	0	.	3.611	0.0301

Modeclus Analysis with the JOIN= option  
A Normal Cluster Surrounded by a Ring Cluster

The MODECLUS Procedure  
R=2.5 METHOD=1

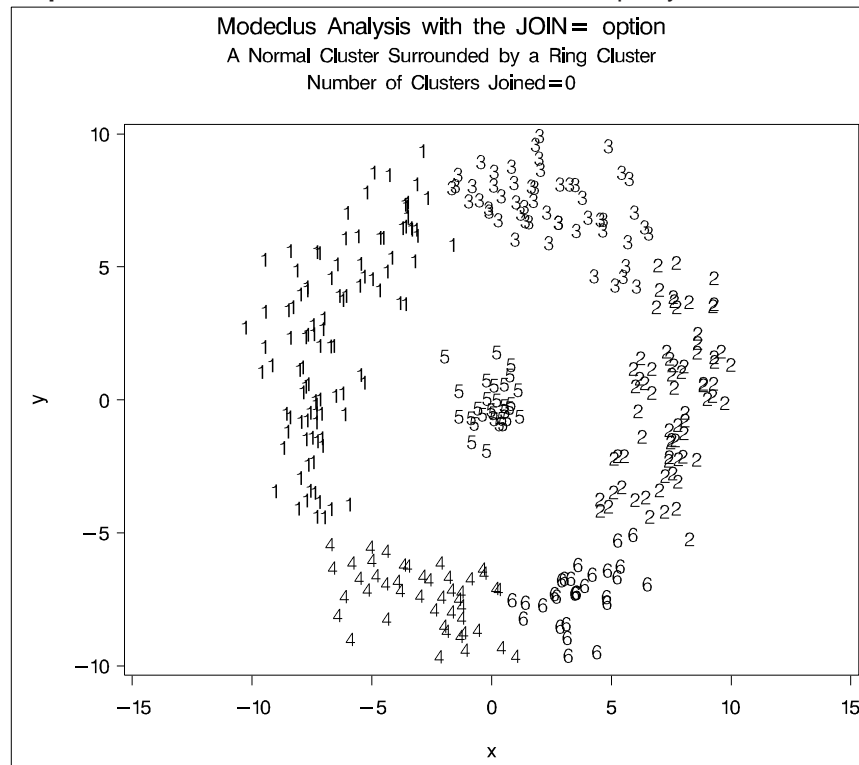
Cluster Statistics									
Cluster	Frequency	Maximum	Boundary	Estimated	-----Saddle Test: Version 92.7-----				
		Estimated Density		Saddle Density	Mode Count	Saddle Count	Overlap Count	Z	Approx P-value
1	300	0.00617328	0	.	39	0	.	4.246	0.0026

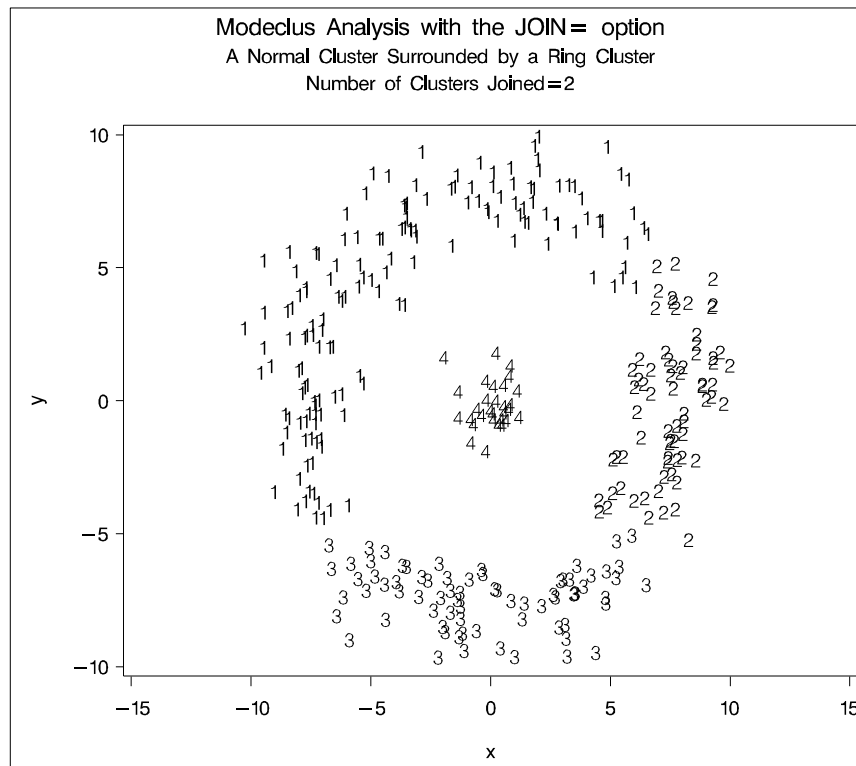
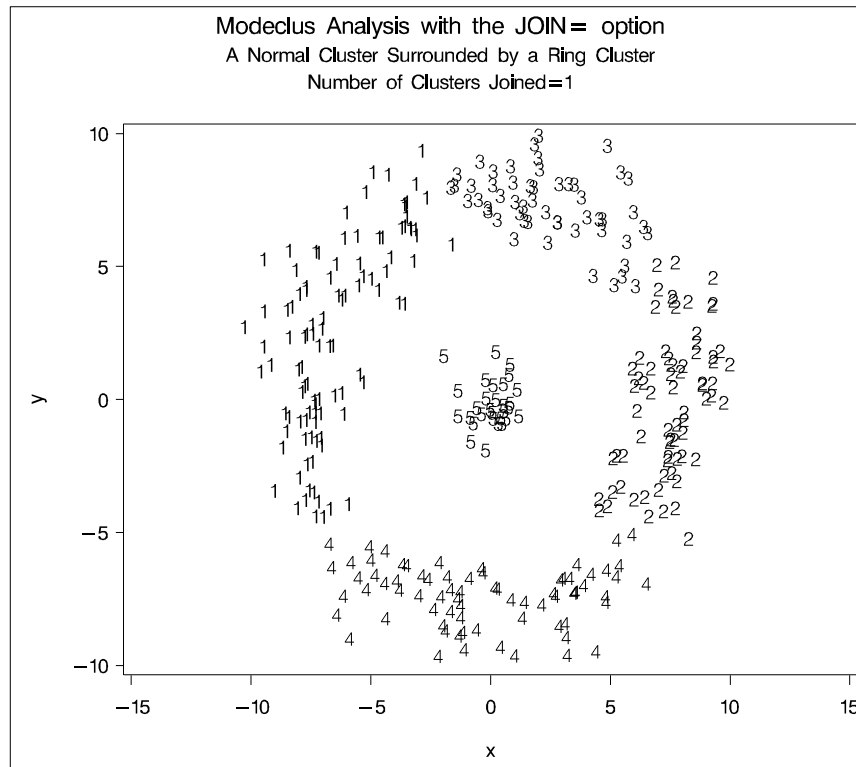
Modeclus Analysis with the JOIN= option  
A Normal Cluster Surrounded by a Ring Cluster

The MODECLUS Procedure

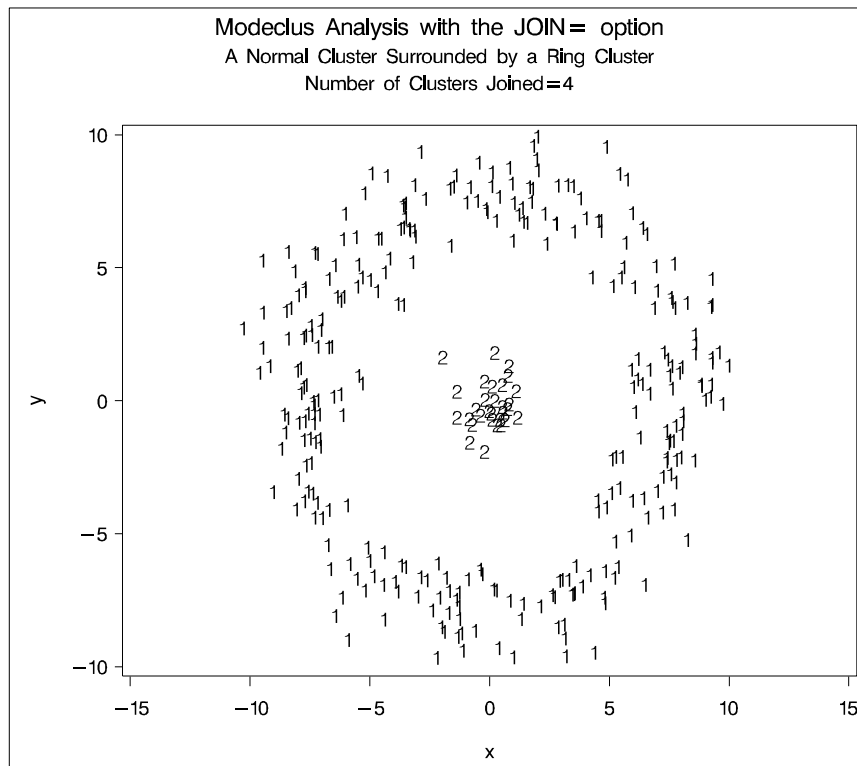
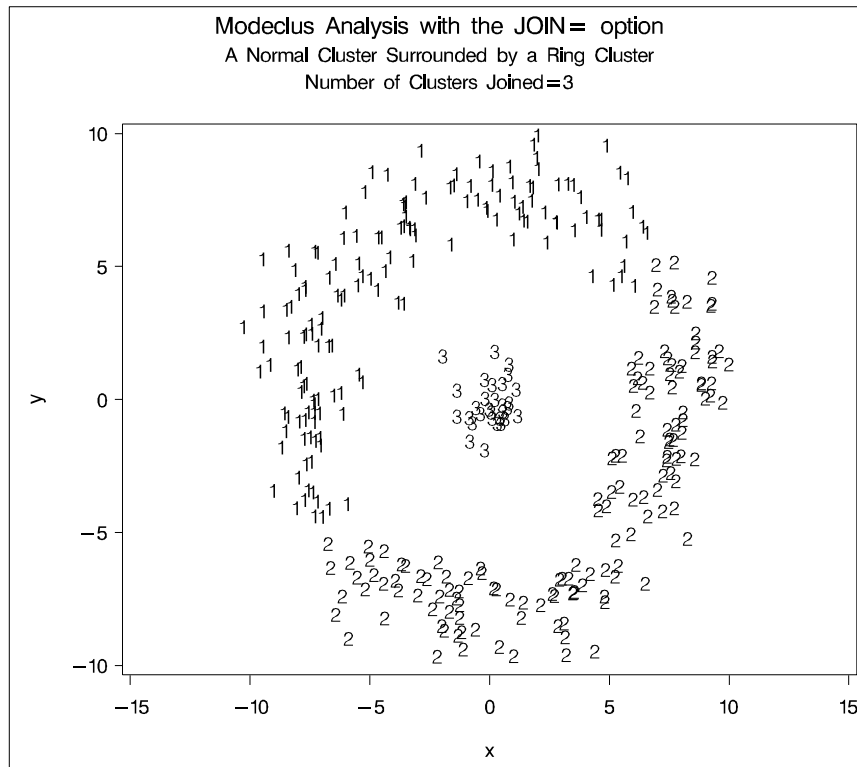
R	Cluster Summary			Frequency of Unclassified Objects
	Number of Clusters Joined	Maximum P-value	Number of Clusters	
2.5	0	1.0000	6	0
2.5	1	0.9999	5	0
2.5	2	0.9990	4	0
2.5	3	0.6447	3	0
2.5	4	0.0301	2	0
2.5	5	0.0026	1	30

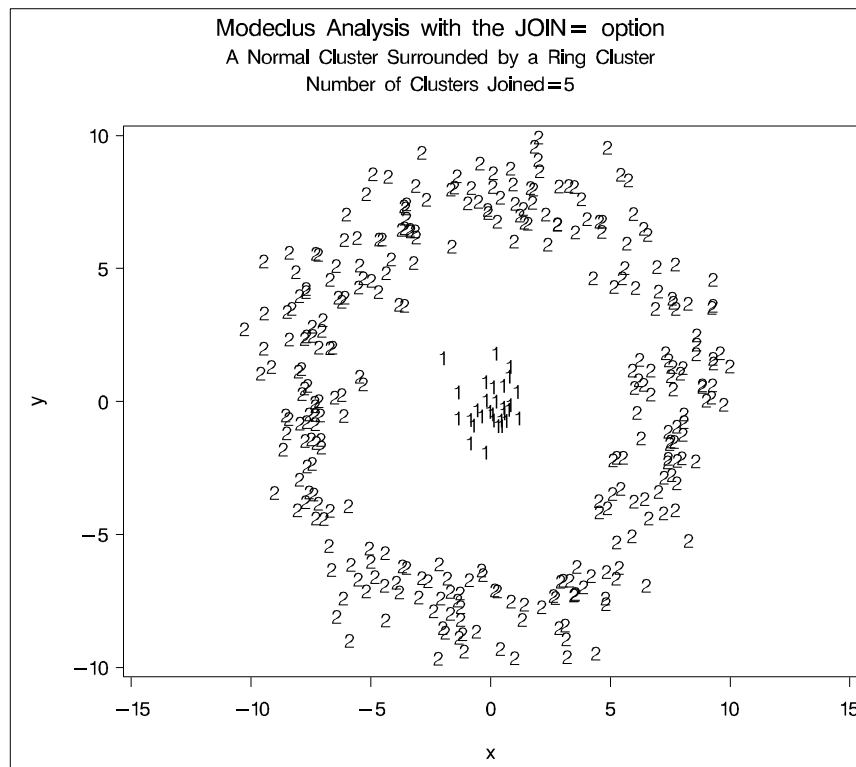
Output 42.3.3. Scatter Plots of Cluster Memberships by \_NJOIN\_











### Example 42.4. Cluster Analysis: Hertzsprung-Russell Plot

This example uses computer-generated data to mimic a Hertzsprung-Russell plot (Struve and Zebergs 1962, p. 259) of the temperature and luminosity of stars. The data are plotted and displayed in Output 42.4.1; see “Example 4 from Proc Modeclus” in the SAS/STAT Sample Program Library for the complete data set. It appears that there are two main groups of stars and a collection of isolated stars. The long straggling group of points appearing diagonally across the figure represents the main group of stars; the more compact group in the top right-hand corner contains giant stars. The JOIN= option is specified at a 0.05 significance level with various smoothing parameters. The CK=5 option is specified in order to prevent the numerous outliers from forming separate clusters. The results from PROC MODECLUS is displayed in Output 42.4.2. The cluster memberships are then plotted by PROC GPLOT, as displayed in Output 42.4.3.

Notice in Output 42.4.3 that the graphic output from PROC GPLOT when `_R_ = 2.5` is not available because only one cluster remains after joining at a 5% significance level, and the results are not written to the OUT= data set. See the description of the JOIN= option on page 2240 for more information.

```

title 'Hertzsprung-Russell Plot of Visible Stars';
title2 'Computer-Generated Fake Data';
data hr;
  input x y @@;
  label x='-Temperature'
        y='-Luminosity';
  datalines;
1.0 12.8 0.9 13.7 0.9 12.9 1.0 12.3 1.0 12.2
2.6 10.9 2.4 10.9 2.5 11.2 2.3 11.5 2.6 12.0
2.4 12.1 2.3 10.9 2.6 11.5 2.5 11.9 2.4 11.0
3.4 11.1 3.3 11.2 3.4 11.1 3.4 9.9 3.2 10.4

... 150 lines omitted ...

18.5 12.6 14.2 16.1 23.2 6.6 11.4 12.4 20.4 11.7
20.9 8.1 18.9 13.7 16.9 9.7 15.5 9.9 18.3 14.2
19.3 13.7 17.0 12.9 10.1 11.6 17.9 13.5 14.3 1.4
13.1 -0.8 8.1 -0.9 20.0 7.0 21.0 8.5 15.6 13.2
;

symbol1 value=circle c=white;
symbol2 value=plus c=yellow;
symbol3 value=triangle c=cyan;
legend1 frame cframe=ligr cborder=black
        position=center value=(justify=center);
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;

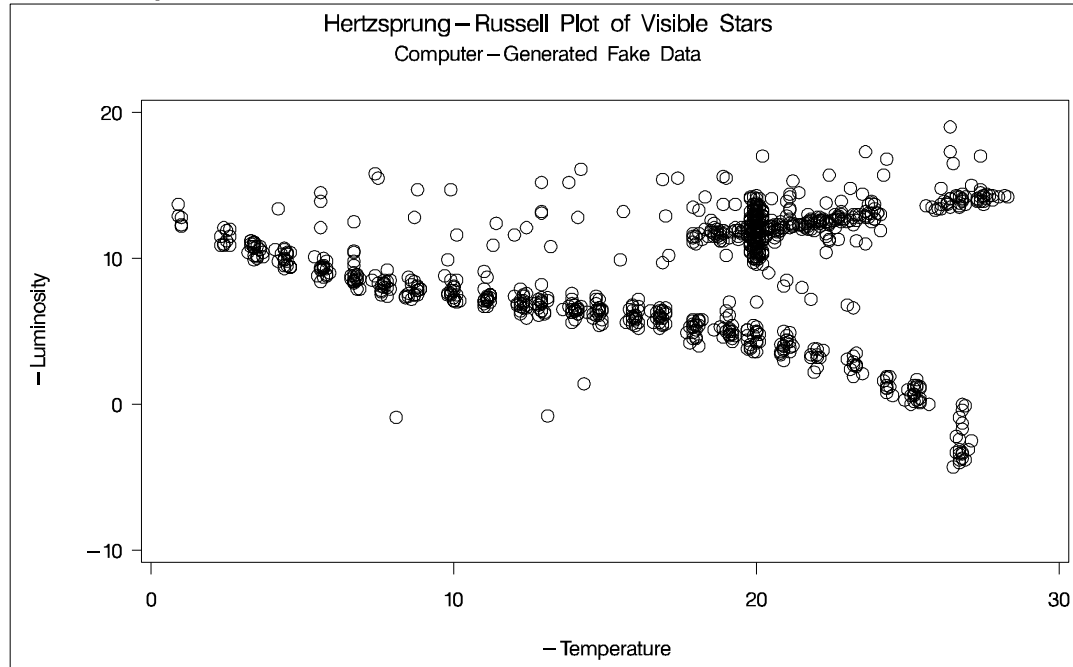
proc gplot;
  plot y*x/legend=legend1 frame cframe=ligr vzero
        vaxis=axis1 haxis=axis2 ;

proc modeclus data=hr m=1 r=1 1.5 2 2.5 ck=5
  join=.05 short out=out;
run;

title2 'MODECLUS Analysis';
proc gplot;
  plot y*x=cluster/frame cframe=ligr
        vzero legend=legend1
        vaxis=axis1 haxis=axis2;

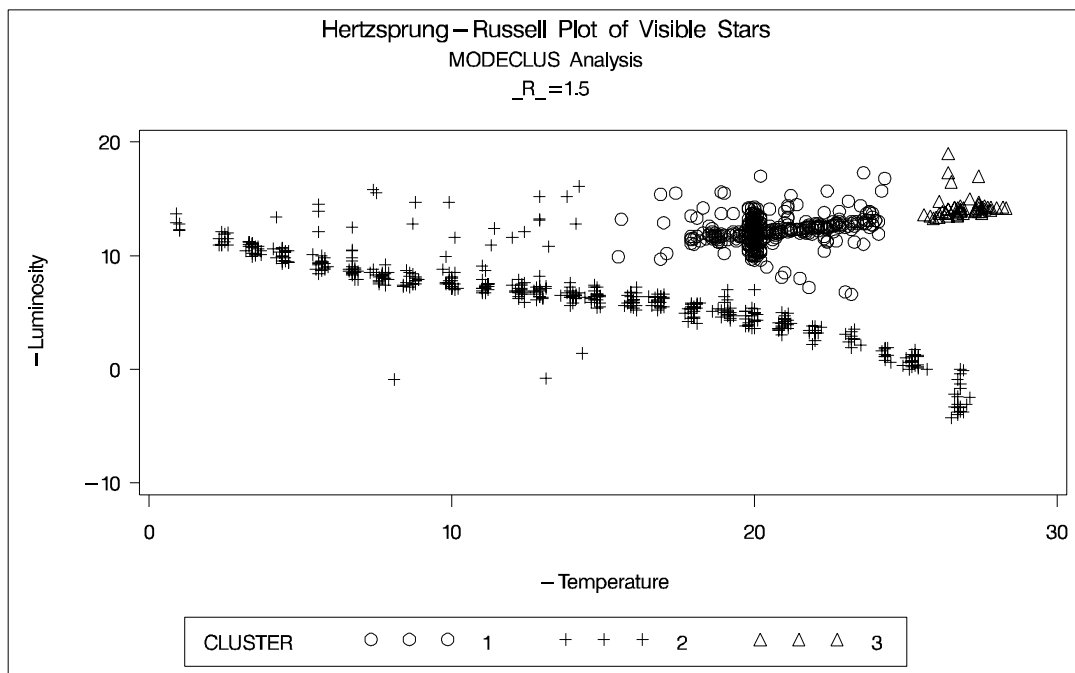
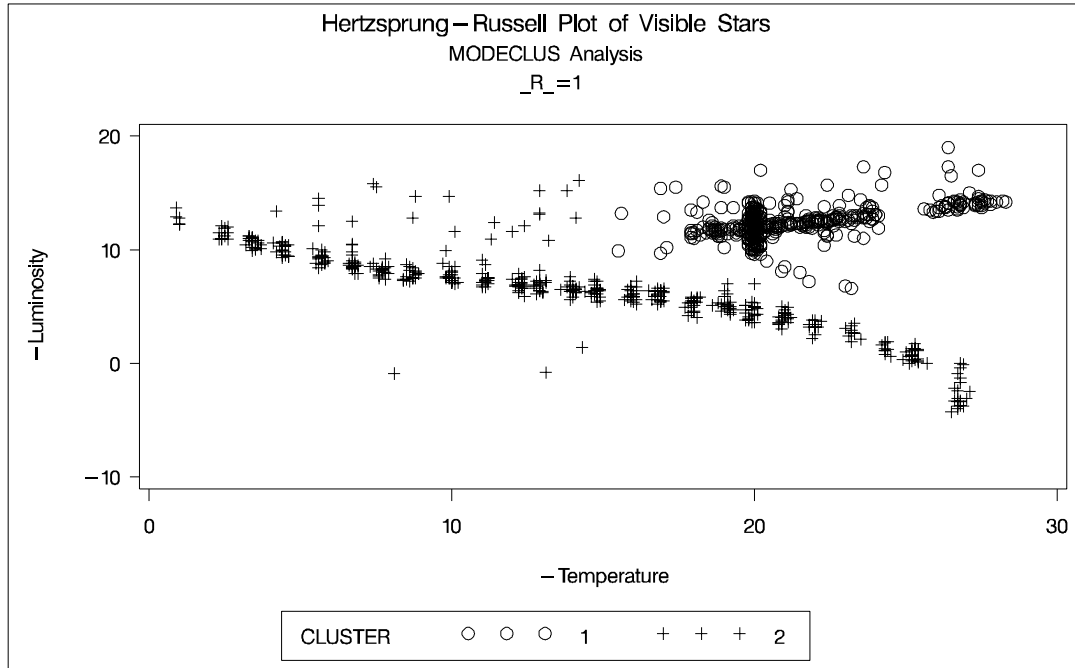
  by _R_;
run;

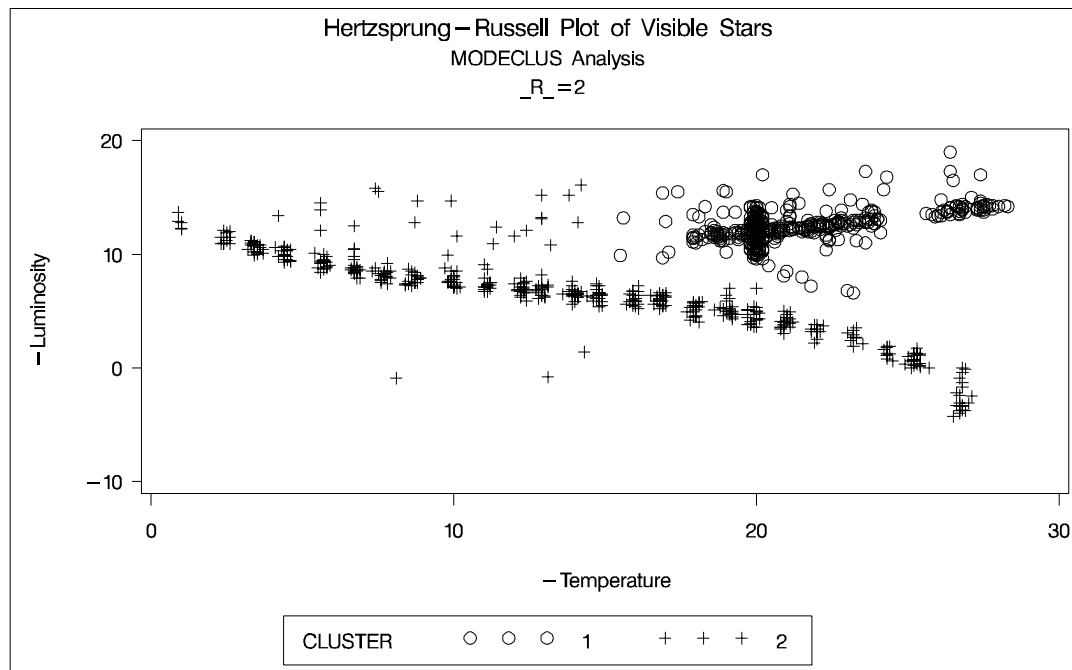
```

**Output 42.4.1.** Scatter Plot of Data**Output 42.4.2.** Results from PROC MODECLUS

Hertzsprung-Russell Plot of Visible Stars Computer-Generated Fake Data					
The MODECLUS Procedure					
Cluster Summary					
R	CK	Number of Clusters Joined	Maximum P-value	Number of Clusters	Frequency of Unclassified Objects
1	5	14	0.0001	2	0
1.5	5	6	0.0000	3	0
2	5	4	0.0000	2	0
2.5	5	2	0.0000	1	0

**Output 42.4.3.** Scatter Plots of Cluster Memberships by `_R_`





### Example 42.5. Using the TRACE Option when METHOD=6

To illustrate how the TRACE option can help you to understand the clustering process when METHOD=6 is specified, the following data set is created with 12 observations.

```
data test;
  input x@@;
  datalines;
1 2 3 4 5 7.5 9 11.5 13 14.5 15 16
;
```

The first five observations seem to be close to each other, and the last five observations seem to be close to each other. Observation 6 is separated from the first five observations with a (Euclidean) distance of 2.5, and the same distance separates observation 7 from the last five observations. Observations 6 and 7 differ by 1.5.

Suppose METHOD=6 with a radius=2.5 is chosen for the cluster analysis. You can specify the TRACE option to understand how each observation is assigned.

The following statements produce Output 42.5.1 and Output 42.5.2:

```

/*-- METHOD=6 with TRACE and THRESHOLD=0.5 (default) --*/
proc modeclus method=6 r=2.5 trace short out=out;
    var x;
run;

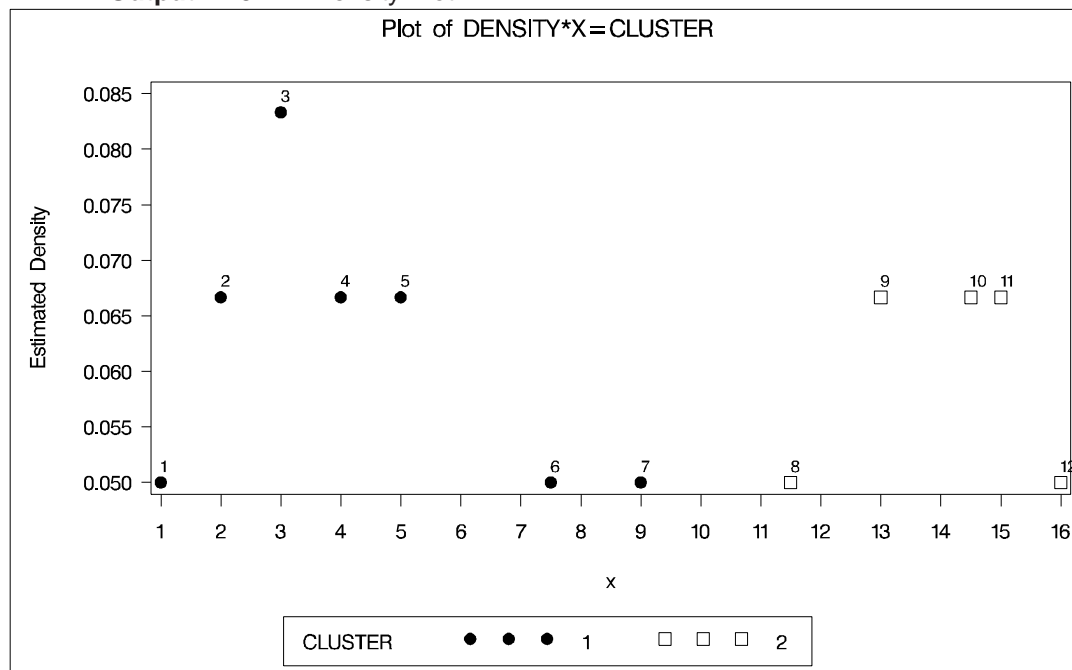
data markobs;
    drop _r_ _method_ _obs_ density cluster;
    length function style $8 text $ 2;
    retain xsys '2' ysys '2' hsys '1' when 'a';
    set out;
    /* create the text for obs */
    function='label'; size=4;
    style='swiss';
    text=left(put(_obs_,2.));
    position='3';
    x=x; y=density;
    output;
run;

legend1 frame cframe=ligr cborder=black
    position=center value=(justify=center);
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;
title 'Plot of DENSITY*X=CLUSTER';
proc gplot data=out;
    plot density*x=cluster/ annotate=markobs
                                frame cframe=ligr
                                legend=legend1
                                vaxis=axis1 haxis=axis2;
run;

```

**Output 42.5.1.** Partial Output of METHOD=6 with TRACE and Default THRESHOLD=

The MODECLUS Procedure				
R=2.5 METHOD=6				
Trace of Clustering Algorithm				
Cluster				
Obs	Density	Old	New	Ratio
3	0.0833333	-1	1	M
2	0.0666667	0	1	N
4	0.0666667	0	1	N
5	0.0666667	0	1	N
1	0.0500000	0	1	N
6	0.0500000	0	1	0.571
7	0.0500000	-1	1	0.500
9	0.0666667	-1	2	M
8	0.0500000	0	2	N
10	0.0666667	-1	2	S
12	0.0500000	0	2	N
11	0.0666667	-1	2	S

**Output 42.5.2.** Density Plot

Notice that in Output 42.5.1, observation 7 is originally a seed (indicated by a value of -1 in the “Old” column) and then assigned to cluster 1. This is because the ratio of observation 7 to cluster 1 is 0.5 and is not less than the default value of THRESHOLD= (0.5).

If the value of the THRESHOLD= option is increased to 0.55, observation 7 should be excluded from cluster 1 and the cluster membership of observation 7 is changed.

The following statements produce Output 42.5.3 and Output 42.5.4:

```

/*-- METHOD=6 with TRACE and THRESHOLD=0.55 --*/
proc modeclus method=6 r=2.5 trace threshold=0.55 short
    out=out;
    var x;
run;

. . . (the Data Step and the PROC GPLOT statement
       are omitted because they are the same as the
       previous job)

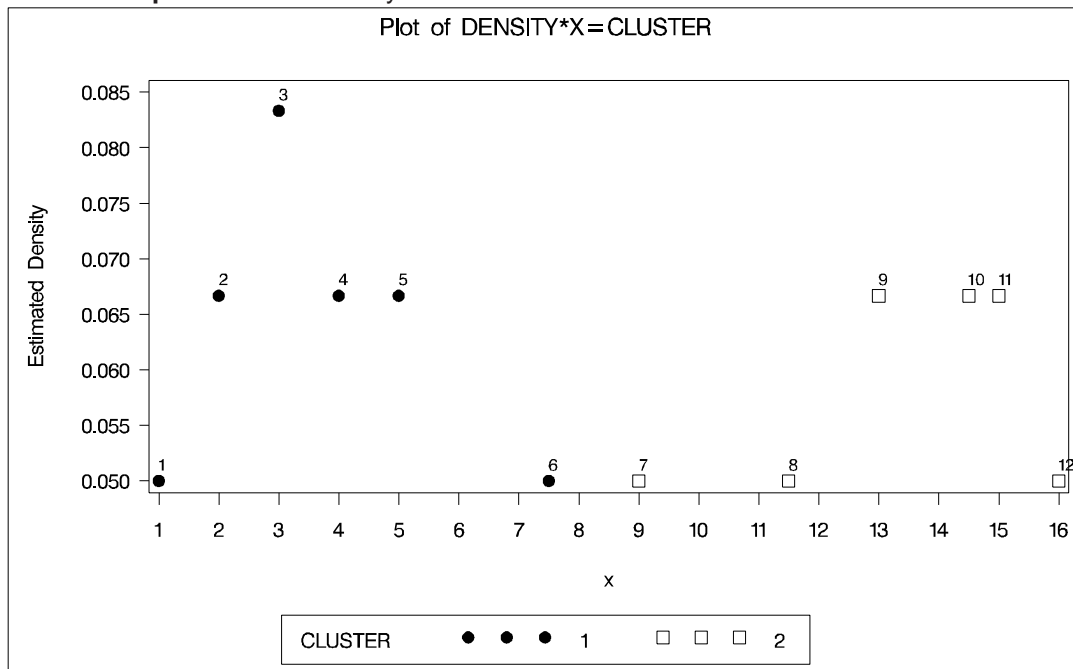
```



**Output 42.5.3.** Partial Output of METHOD=6 with TRACE and THRESHOLD=.55

The MODECLUS Procedure				
R=2.5 METHOD=6				
Trace of Clustering Algorithm				
Cluster				
Obs	Density	Old	New	Ratio
3	0.0833333	-1	1	M
2	0.0666667	0	1	N
4	0.0666667	0	1	N
5	0.0666667	0	1	N
1	0.0500000	0	1	N
6	0.0500000	0	1	0.571
9	0.0666667	-1	2	M
8	0.0500000	0	2	N
10	0.0666667	-1	2	S
12	0.0500000	0	2	N
11	0.0666667	-1	2	S
7	0.0500000	-1	2	S

**Output 42.5.4.** Density Plot



In Output 42.5.3, observation 7 is a seed that is excluded by cluster 1 because its ratio to cluster 1 is less than 0.55. Being a neighbor of a member (observation 8) of cluster 2, observation 7 eventually joins cluster 2 even though it remains a “SEED.” (See **Step 2.2** in the section “METHOD=6” on page 2250.)

## References

---

- Barnett, V., ed. (1981), *Interpreting Multivariate Data*, New York: John Wiley & Sons, Inc.
- Gitman, I. (1973), "An Algorithm for Nonsupervised Pattern Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, 66–74.
- Hartigan, J.A. and Hartigan, P.M. (1985), "The Dip Test of Unimodality," *Annals of Statistics*, 13, 70–84.
- Huizinga, D.H. (1978), "A Natural or Mode Seeking Cluster Analysis Algorithm," Technical Report 78-1, Behavioral Research Institute, 2305 Canyon Blvd., Boulder, Colorado 80302.
- Koontz, W.L.G. and Fukunaga, K. (1972a), "A Nonparametric Valley-Seeking Technique for Cluster Analysis," *IEEE Transactions on Computers*, C-21, 171–178.
- Koontz, W.L.G. and Fukunaga, K. (1972b), "Asymptotic Analysis of a Nonparametric Clustering Technique," *IEEE Transactions on Computers*, C-21, 967–974.
- Koontz, W.L.G., Narendra, P.M., and Fukunaga, K. (1976), "A Graph-Theoretic Approach to Nonparametric Cluster Analysis," *IEEE Transactions on Computers*, C-25, 936–944.
- Minnotte, M.C. (1992), "A Test of Mode Existence with Applications to Multimodality," Ph.D. thesis, Rice University, Department of Statistics.
- Mizoguchi, R. and Shimura, M. (1980), "A Nonparametric Algorithm for Detecting Clusters Using Hierarchical Structure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2, 292–300.
- Müller, D.W. and Sawitzki, G. (1991), "Excess Mass Estimates and Tests for Multimodality," *JASA* 86, 738–746.
- Polonik, W. (1993), "Measuring Mass Concentrations and Estimating Density Contour Clusters—An Excess Mass Approach," Technical Report, Beitrage zur Statistik Nr. 7, Universitaet Heidelberg.
- Sarle, W.S. (1982), "Cluster Analysis by Least Squares," *SAS Users Group International Conference Proceedings: SUGI 7*, 651–653.
- Scott, D.W. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York: John Wiley & Sons, Inc.
- Silverman, B.W. (1986), *Density Estimation*, New York: Chapman and Hall.
- Struve, O. and Zebergs, V. (1962), *Astronomy of the Twentieth Century*, New York: Macmillan.
- Tukey, P.A. and Tukey, J.W. (1981), "Data-Driven View Selection; Agglomeration and Sharpening," in Barnett (1981).
- Wong, M.A. and Lane, T. (1983), "A  $k$ th Nearest Neighbor Clustering Procedure," *Journal of the Royal Statistical Society, Series B*, 45, 362–368.

- Wong, M.A. and Schaack, C. (1982), "Using the  $k$ th Nearest Neighbor Clustering Procedure to Determine the Number of Subpopulations," *American Statistical Association 1982 Proceedings of the Statistical Computing Section*, 40–48.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/STAT® User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999.

**SAS/STAT® User's Guide, Version 8**

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-494-2

All rights reserved. Produced in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

**U.S. Government Restricted Rights Notice.** Use, duplication, or disclosure of the software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.